Network Working Group
Request for Comments: 3761
Obsoletes: 2916
Category: Standards Track

P. Faltstrom
Cisco Systems, Inc.
M. Mealling
VeriSign

April 2004

The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)

Status of this Memo

Copyright Notice

Abstract

This document discusses the use of the Domain Name System (DNS) for storage of E.164 numbers. More specifically, how DNS can be used for identifying available services connected to one E.164 number. It specifically obsoletes RFC 2916 to bring it in line with the Dynamic Delegation Discovery System (DDDS) Application specification found in the document series specified in RFC 3401. It is very important to note that it is impossible to read and understand this document without reading the documents discussed in RFC 3401.

Table of Contents

1. Introduction

This document discusses the use of the Domain Name System (DNS) for storage of E.164 numbers. More specifically, how DNS can be used for identifying available services connected to one E.164 number. It specifically obsoletes RFC 2916 to bring it in line with the Dynamic Delegation Discovery System (DDDS) Application specification found in the document series specified in RFC 3401 [6]. It is very important to note that it is impossible to read and understand this document without reading the documents discussed in RFC 3401 [6].

Through transformation of International Public Telecommunication Numbers in the international format [5], called within this document E.164 numbers, into DNS names and the use of existing DNS services like delegation through NS records and NAPTR records, one can look up what services are available for a specific E.164 in a decentralized way with distributed management of the different levels in the lookup process.

The domain "e164.arpa" is being populated in order to provide the infrastructure in DNS for storage of E.164 numbers. In order to facilitate distributed operations, this domain is divided into subdomains. Holders of E.164 numbers which want to be listed in DNS should contact the appropriate zone administrator according to the

policy which is attached to the zone. One should start looking for this information by examining the SOA resource record associated with the zone, just like in normal DNS operations.

Of course, as with other domains, policies for such listings will be controlled on a subdomain basis and may differ in different parts of the world.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [1].

All other capitalized terms are taken from the vocabulary found in the DDDS algorithm specification found in RFC 3403 [2].

1.2. Use for these mechanisms for private dialing plans

This document describes the operation of these mechanisms in the context of numbers allocated according to the ITU-T recommendation E.164. The same mechanisms might be used for private dialing plans. If these mechanisms are re-used, the suffix used for the private dialing plan MUST NOT be e164.arpa, to avoid conflict with this specification. Parties to the private dialing plan will need to know the suffix used by their private dialing plan for correct operation of these mechanisms. Further, the application unique string used SHOULD be the full number as specified, but without the leading '+', and such private use MUST NOT be called "ENUM".

1.3. Application of local policy

The Order field in the NAPTR record specifies in what order the DNS records are to be interpreted. This is because DNS does not guarantee the order of records returned in the answer section of a DNS packet. In most ENUM cases this isn't an issue because the typical regular expression will be '!^.*$!' since the first query often results in a terminal Rule.

But there are other cases (non-terminal Rules) where two different Rules both match the given Application Unique String. As each Rule is evaluated within the algorithm, one may match a more significant piece of the AUS than the other. For example, by using a non-terminal NAPTR a given set of numbers is sent to some private-dialing-plan-specific zone. Within that zone there are two Rules that state that if a match is for the entire exchange and the service is SIP related then the first, SIP-specific rule is used. But the other Rule matches a longer piece of the AUS, specifying that for

some other service (instant messaging) that the Rule denotes a departmental level service. If the shorter matching Rule comes before the longer match, it can 'mask' the other rules. Thus, the order in which each Rule is tested against the AUS is an important corner case that many DDDS applications take advantage of.

In the case where the zone authority wishes to state that two Rules have the same effect or are identical in usage, then the Order for those records is set to the same value. In that case, the Preference is used to specify a locally over-ridable suggestion by the zone authority that one Rule might simply be better than another for some reason.

For ENUM this specifies where a client is allowed to apply local policy and where it is not. The Order field in the NAPTR is a request from the holder of the E.164 number that the records be handled in a specific way. The Preference field is merely a suggestion from that E.164 holder that one record might be better than another. A client implementing ENUM MUST adhere to the Order field but can simply take the Preference value "on advisement" as part of a client context specific selection method.

2. The ENUM Application Specifications

This template defines the ENUM DDDS Application according to the rules and requirements found in [7]. The DDDS database used by this Application is found in [2] which is the document that defines the NAPTR DNS Resource Record type.

ENUM is only applicable for E.164 numbers. ENUM compliant applications MUST only query DNS for what it believes is an E.164 number. Since there are numerous dialing plans which can change over time, it is probably impossible for a client application to have perfect knowledge about every valid and dialable E.164 number. Therefore a client application, doing everything within its power, can end up with what it thinks is a syntactically correct E.164 number which in reality is not actually valid or dialable. This implies that applications MAY send DNS queries when, for example, a user mistypes a number in a user interface. Because of this, there is the risk that collisions between E.164 numbers and non-E.164 numbers can occur. To mitigate this risk, the E2U portion of the service field MUST NOT be used for non-E.164 numbers.

2.1. Application Unique String

The Application Unique String is a fully qualified E.164 number minus any non-digit characters except for the '+' character which appears at the beginning of the number. The "+" is kept to provide a well understood anchor for the AUS in order to distinguish it from other telephone numbers that are not part of the E.164 namespace.

For example, the E.164 number could start out as "+44-116-496-0348". To ensure that no syntactic sugar is allowed into the AUS, all non-digits except for "+" are removed, yielding "+441164960348".

2.2. First Well Known Rule

The First Well Known Rule for this Application is the identity rule. The output of this rule is the same as the input. This is because the E.164 namespace and this Applications databases are organized in such a way that it is possible to go directly from the name to the smallest granularity of the namespace directly from the name itself.

Take the previous example, the AUS is "+441164960348". Applying the First Well Known Rule produces the exact same string, "+441164960348".

2.3. Expected Output

The output of the last DDDS loop is a Uniform Resource Identifier in its absolute form according to the 'absoluteURI' production in the Collected ABNF found in RFC2396 [4].

2.4. Valid Databases

At present only one DDDS Database is specified for this Application. "Dynamic Delegation Discovery System (DDDS) Part Three: The DNS Database" (RFC 3403) [2] specifies a DDDS Database that uses the NAPTR DNS resource record to contain the rewrite rules. The Keys for this database are encoded as domain-names.

The output of the First Well Known Rule for the ENUM Application is the E.164 number minus all non-digit characters except for the +. In order to convert this to a unique key in this Database the string is converted into a domain-name according to this algorithm:

1. Remove all characters with the exception of the digits. For example, the First Well Known Rule produced the Key "+442079460148". This step would simply remove the leading "+", producing "442079460148".

2. Put dots (".") between each digit. Example:
   4.4.2.0.7.9.4.6.0.1.4.8

3. Reverse the order of the digits. Example:
   8.4.1.0.6.4.9.7.0.2.4.4

4. Append the string ".e164.arpa" to the end. Example:
   8.4.1.0.6.4.9.7.0.2.4.4.e164.arpa

This domain-name is used to request NAPTR records which may contain the end result or, if the flags field is blank, produces new keys in the form of domain-names from the DNS.

Some nameserver implementations attempt to be intelligent about items that are inserted into the additional information section of a given DNS response. For example, BIND will attempt to determine if it is authoritative for a domain whenever it encodes one into a packet. If it is, then it will insert any A records it finds for that domain into the additional information section of the answer until the packet reaches the maximum length allowed. It is therefore potentially useful for a client to check for this additional information. It is also easy to contemplate an ENUM enhanced nameserver that understand the actual contents of the NAPTR records it is serving and inserts more appropriate information into the additional information section of the response. Thus, DNS servers MAY interpret Flag values and use that information to include appropriate resource records in the Additional Information portion of the DNS packet. Clients are encouraged to check for additional information but are not required to do so. See the Additional Information Processing section of RFC 3403 [2], Section 4.2 for more information on NAPTR records and the Additional Information section of a DNS response packet.

The character set used to encode the substitution expression is UTF-8. The allowed input characters are all those characters that are allowed anywhere in an E.164 number. The characters allowed to be in a Key are those that are currently defined for DNS domain-names.

2.4.1. Flags

This Database contains a field that contains flags that signal when the DDDS algorithm has finished. At this time only one flag, "U", is defined. This means that this Rule is the last one and that the output of the Rule is a URI [4]. See RFC 3404 [3].

If a client encounters a record with an unknown flag, it MUST ignore it and move to the next Rule. This test takes precedence over any ordering since flags can control the interpretation placed on fields.

A novel flag might change the interpretation of the regexp and/or replacement fields such that it is impossible to determine if a record matched a given target.

If this flag is not present then this rule is non-terminal. If a Rule is non-terminal then clients MUST use the Key produced by this Rewrite Rule as the new Key in the DDDS loop (i.e., causing the client to query for new NAPTR records at the domain-name that is the result of this Rule).

2.4.2. Services Parameters

Service Parameters for this Application take the following form and are found in the Service field of the NAPTR record.

```
service-field = "E2U" 1*(servicespec)
servicespec   = "+" enumservice
enumservice   = type 0*(subtypespec)
subtypespec   = ":" subtype
type          = 1*32(ALPHA / DIGIT)
subtype       = 1*32(ALPHA / DIGIT)
```

In other words, a non-optional "E2U" (used to denote ENUM only Rewrite Rules in order to mitigate record collisions) followed by 1 or more or more Enumservices which indicate what class of functionality a given end point offers. Each Enumservice is indicated by an initial '+' character.

2.4.2.1. ENUM Services

Enumservice specifications contain the functional specification (i.e., what it can be used for), the valid protocols, and the URI schemes that may be returned. Note that there is no implicit mapping between the textual string "type" or "subtype" in the grammar for the Enumservice and URI schemes or protocols. The mapping, if any, must be made explicit in the specification for the Enumservice itself. A registration of a specific Type also has to specify the Subtypes allowed.

The only exception to the registration rule is for Types and Subtypes used for experimental purposes, and those are to start with the facet "X-". These elements are unregistered, experimental, and should be used only with the active agreement of the parties exchanging them.

The registration mechanism is specified in Section 3.

2.5. What constitutes an 'Enum Resolver'?

There has been some confusion over what exactly an ENUM Resolver returns and what relation that has to the 'Note 1' section in RFC 3402. On first reading it seems as though it might be possible for an ENUM Resolver to return two Rules.

The ENUM algorithm always returns a single rule. Specific applications may have application-specific knowledge or facilities that allow them to present multiple results or speed selection, but these should never change the operation of the algorithm.

3. Registration mechanism for Enumservices

As specified in the ABNF found in Section 2.4.2, an 'enumservice' is made up of 'types' and 'subtypes'. For any given 'type', the allowable 'subtypes' must be specified in the registration. There is currently no concept of a registered 'subtype' outside the scope of a given 'type'. Thus the registration process uses the 'type' as its main key within the IANA Registry. While the combination of each type and all of its subtypes constitutes the allowed values for the 'enumservice' field, it is not sufficient to simply document those values. A complete registration will also include the allowed URI schemes, a functional specification, security considerations, intended usage, and any other information needed to allow for interoperability within ENUM. In order to be a registered ENUM Service, the entire specification, including the template, requires approval by the IESG and publication of the Enumservice registration specification as an RFC.

3.1. Registration Requirements

Service registration proposals are all expected to conform to various requirements laid out in the following sections.

3.1.1. Functionality Requirement

A registered Enumservice must be able to function as a selection mechanism when choosing one NAPTR resource record from another. That means that the registration MUST specify what is expected when using that very NAPTR record, and the URI which is the outcome of the use of it.

Specifically, a registered Enumservice MUST specify the URI scheme(s) that may be used for the Enumservice, and, when needed, other information which will have to be transferred into the URI resolution process itself (LDAP Distinguished Names, transferring of the AUS into the resulting URI, etc).

3.1.2. Naming requirement

An Enumservice MUST be unique in order to be useful as a selection criteria. Since an Enumservice is made up of a type and a type-dependent subtype, it is sufficient to require that the 'type' itself be unique. The 'type' MUST be unique, conform to the ABNF specified in Section 2.4.2, and MUST NOT start with the facet "X-" which is reserved for experimental, private use.

The subtype, being dependent on the type, MUST be unique within a given 'type'. It must conform to the ABNF specified in Section 2.4.2, and MUST NOT start with the facet "X-" which is reserved for experimental, private use. The subtype for one type MAY be the same as a subtype for a different registered type but it is not sufficient to simply reference another type's subtype. The function of each subtype must be specified in the context of the type being registered.

3.1.3. Security requirement

An analysis of security issues is required for all registered Enumservices. (This is in accordance with the basic requirements for all IETF protocols.)

All descriptions of security issues must be as accurate as possible regardless of registration tree. In particular, a statement that there are "no security issues associated with this Enumservice" must not be confused with "the security issues associated with this Enumservice have not been assessed".

There is no requirement that an Enumservice must be secure or completely free from risks. Nevertheless, all known security risks must be identified in the registration of an Enumservice.

The security considerations section of all registrations is subject to continuing evaluation and modification.

Some of the issues that should be looked at in a security analysis of an Enumservice are:

1. Complex Enumservices may include provisions for directives that institute actions on a user's resources. In many cases provision can be made to specify arbitrary actions in an unrestricted fashion which may then have devastating results. Especially if there is a risk for a new ENUM lookup, and because of that an infinite loop in the overall resolution process of the E.164.

2. Complex Enumservices may include provisions for directives that institute actions which, while not directly harmful, may result in disclosure of information that either facilitates a subsequent attack or else violates the users privacy in some way.

3. An Enumservice might be targeted for applications that require some sort of security assurance but do not provide the necessary security mechanisms themselves. For example, an Enumservice could be defined for storage of confidential security services information such as alarm systems or message service passcodes, which in turn require an external confidentiality service.

3.1.4. Publication Requirements

Proposals for Enumservices registrations MUST be published as one of the following documents; RFC on the Standards Track, Experimental RFC, or as a BCP.

IANA will retain copies of all Enumservice registration proposals and "publish" them as part of the Enumservice Registration tree itself.

3.2. Registration procedure

3.2.1. IANA Registration

Provided that the Enumservice has obtained the necessary approval, and the RFC is published, IANA will register the Enumservice and make the Enumservice registration available to the community in addition to the RFC publication itself.

3.2.1.1. Location of Enumservice Registrations

Enumservice registrations will be published in the IANA repository and made available via anonymous FTP at the following URI: "ftp://ftp.iana.org/assignments/enum-services/".

3.2.1.2. Change Control

Change control of Enumservices stay with the IETF via the RFC publication process. Especially, Enumservice registrations may not be deleted; Enumservices which are no longer believed appropriate for use can be declared OBSOLETE by publication of a new RFC and a change to their "intended use" field; such Enumservice will be clearly marked in the lists published by IANA.

3.2.2. Registration Template

Enumservice Type:

Enumservice Subtype(s):

URI Scheme(s):

Functional Specification:

Security considerations:

Intended usage: (One of COMMON, LIMITED USE or OBSOLETE)

Author:

Any other information that the author deems interesting:

Note: In the case where a particular field has no value, that field is left completely blank, especially in the case where a given type has no subtypes.

4. Examples

The examples below use theoretical services that contain Enumservices which might not make sense, but that are still used for educational purposes. For example, the protocol used is in some cases exactly the same string as the URI scheme. That was the specification in RFC 2916, but this 'default' specification of an Enumservice is no longer allowed. All Enumservices need to be registered explicitly by the procedure specified in section Section 3.

4.1. Example

```
$ORIGIN 3.8.0.0.6.9.2.3.6.1.4.4.e164.arpa.
   NAPTR 10 100 "u" "E2U+sip" "!^.*$!sip:info@example.com!" .
   NAPTR 10 101 "u" "E2U+h323" "!^.*$!h323:info@example.com!" .
   NAPTR 10 102 "u" "E2U+msg" "!^.*$!mailto:info@example.com!" .
```

This describes that the domain 3.8.0.0.6.9.2.3.6.1.4.4.e164.arpa. is preferably contacted by SIP, secondly via H.323 for voice, and thirdly by SMTP for messaging. Note that the tokens "sip", "h323", and "msg" are Types registered with IANA, and they have no implicit connection with the protocols or URI schemes with the same names.

In all cases, the next step in the resolution process is to use the resolution mechanism for each of the protocols, (specified by the URI schemes sip, h323 and mailto) to know what node to contact for each.

5. IANA Considerations

RFC 2916 (which this document replaces) requested IANA to delegate the E164.ARPA domain following instructions to be provided by the IAB. The domain was delegated according to those instructions. Names within this zone are to be delegated to parties according to the ITU-T Recommendation E.164. The names allocated should be hierarchic in accordance with ITU-T Recommendation E.164, and the codes should be assigned in accordance with that Recommendation.

IAB is to coordinate with ITU-T TSB if the technical contact for the domain e164.arpa is to change, as ITU-T TSB has an operational working relationship with this technical contact which needs to be reestablished.

Delegations in the zone e164.arpa (not delegations in delegated domains of e164.arpa) should be done after Expert Review, and the IESG will appoint a designated expert.

IANA has created a registry for Enumservices as specified in Section 3. Whenever a new Enumservice is registered by the RFC process in the IETF, IANA is at the time of publication of the RFC to register the Enumservice and add a pointer to the RFC itself.

6. Security Considerations

6.1. DNS Security

As ENUM uses DNS, which in its current form is an insecure protocol, there is no mechanism for ensuring that the data one gets back is authentic. As ENUM is deployed on the global Internet, it is expected to be a popular target for various kind of attacks, and attacking the underlying DNS infrastructure is one way of attacking the ENUM service itself.

There are multiple types of attacks that can happen against DNS that ENUM implementations should be aware of. The following threats are taken from Threat Analysis Of The Domain Name System [10]:

Packet Interception
: Some of the simplest threats against DNS are various forms of packet interception: monkey-in-the-middle attacks, eavesdropping on requests combined with spoofed responses that beat the real response back to the resolver, and so forth. In any of these

scenarios, the attacker can simply tell either party (usually the resolver) whatever it wants that party to believe. While packet interception attacks are far from unique to DNS, DNS's usual behavior of sending an entire query or response in a single unsigned, unencrypted UDP packet makes these attacks particularly easy for any bad guy with the ability to intercept packets on a shared or transit network.

ID Guessing and Query Prediction
Since the ID field in the DNS header is only a 16-bit field and the server UDP port associated with DNS is a well-known value, there are only 2**32 possible combinations of ID and client UDP port for a given client and server. Thus it is possible for a reasonable brute force attack to allow an attacker to masquerade as a trusted server. In most respects, this attack is similar to a packet interception attack except that it does not require the attacker to be on a transit or shared network.

Name-based Attacks
Name-based attacks use the actual DNS caching behavior as a tool to insert bad data into a victim's cache, thus potentially subverting subsequent decisions based on DNS names. Most examples occur with CNAME, NS and DNAME Resource Records as they redirect a victim's query to another location. The common thread in all of these attacks is that response messages allow the attacker to introduce arbitrary DNS names of the attacker's choosing and provide further information that the attacker claims is associated with those names; unless the victim has better knowledge of the data associated with those names, the victim is going to have a hard time defending against this class of attacks.

Betrayal By A Trusted Server
Another variation on the packet interception attack is the trusted server that turns out not to be so trustworthy, whether by accident or by intent. Many client machines are only configured with stub resolvers, and use trusted servers to perform all of their DNS queries on their behalf. In many cases the trusted server is furnished by the user's ISP and advertised to the client via DHCP or PPP options. Besides accidental betrayal of this trust relationship (via server bugs, successful server break-ins, etc), the server itself may be configured to give back answers that are not what the user would expect (whether in an honest attempt to help the user or to further some other goal such as furthering a business partnership between the ISP and some third party).

Denial of Service
: As with any network service (or, indeed, almost any service of any kind in any domain of discourse), DNS is vulnerable to denial of service attacks. DNS servers are also at risk of being used as denial of service amplifiers, since DNS response packets tend to be significantly longer than DNS query packets.

Authenticated Denial of Domain Names
: The existence of RR types whose absence causes an action other than immediate failure (such as missing MX and SRV RRs, which fail over to A RRs) constitutes a real threat. In the specific case of ENUM, even the immediate failure of a missing RR can be considered a problem as a method for changing call routing policy.

Because of these threats, a deployed ENUM service SHOULD include mechanisms which ameliorate these threats. Most of these threats can be solved by verifying the authenticity of the data via mechanisms such as DNSSEC [8] once it is deployed. Others, such and Denial Of Service attacks, cannot be solved by data authentication. It is important to remember that these threats include not only the NAPTR lookups themselves, but also the various records needed for the services to be useful (for example NS, MX, SRV and A records).

Even if DNSSEC is deployed, a service that uses ENUM for address translation should not blindly trust that the peer is the intended party as all kind of attacks against DNS can not be protected against with DNSSEC. A service should always authenticate the peers as part of the setup process for the service itself and never blindly trust any kind of addressing mechanism.

Finally, as an ENUM service will be implementing some type of security mechanism, software which implements ENUM MUST be prepared to receive DNSSEC and other standardized DNS security responses, including large responses, EDNS0 signaling, unknown RRs, etc.

6.2. Caching Security

The caching in DNS can make the propagation time for a change take the same amount of time as the time to live for the NAPTR records in the zone that is changed. The use of this in an environment where IP-addresses are for hire (for example, when using DHCP [9]) must therefore be done very carefully.

6.3. Call Routing Security

There are a number of countries (and other numbering environments) in which there are multiple providers of call routing and number/name-translation services. In these areas, any system that permits users,

or putative agents for users, to change routing or supplier information may provide incentives for changes that are actually unauthorized (and, in some cases, for denial of legitimate change requests). Such environments should be designed with adequate mechanisms for identification and authentication of those requesting changes and for authorization of those changes.

6.4. URI Resolution Security

A large amount of Security Issues have to do with the resolution process itself, and use of the URIs produced by the DDDS mechanism. Those have to be specified in the registration of the Enumservice used, as specified in Section 3.1.3.

7. Acknowledgements

Support and ideas leading to RFC 2916 have come from people at Ericsson, Bjorn Larsson and the group which implemented this scheme in their lab to see that it worked. Input has also arrived from ITU-T SG2, Working Party 1/2 (Numbering, Routing, Global Mobility and Enumservice Definition), the ENUM working group in the IETF, John Klensin and Leif Sunnegardh.

This update of RFC 2916 is created with specific input from: Randy Bush, David Conrad, Richard Hill, Jon Peterson, Jim Reid, Joakim Stralmark, Robert Walter and James Yu.

8. Changes since RFC 2916

Part from clarifications in the text in this document, the major changes are two:

The document uses an explicit DDDS algorithm, and not only NAPTR resource records in an "ad-hoc" mode. In reality this doesn’t imply any changes in deployed base of applications, as the algorithm used for ENUM resolution is exactly the same.

The format of the service field has changed. The old format was of the form "example+E2U", while the new format is "E2U+example". Reason for this change have to with the added subtypes in the enumservice, the ability to support more than one enumservice per NAPTR RR, and a general agreement in the IETF that the main selector between different NAPTR with the same owner (E2U in this case) should be first.

9. References

9.1. Normative References

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

[2] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database", RFC 3403, October 2002.

[3] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Four: The Uniform Resource Identifiers (URI) Resolution Application", RFC 3404, October 2002.

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

[5] ITU-T, "The International Public Telecommunication Number Plan", Recommendation E.164, May 1997.

[6] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part One: The Comprehensive DDDS", RFC 3401, October 2002.

[7] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Two: The Algorithm", RFC 3402, October 2002.

9.2. Informative References

[8] Eastlake, D., "Domain Name System Security Extensions", RFC 2535, March 1999.

[9] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[10] Atkins, D. and R. Austein, "Threat Analysis Of The Domain Name System", Work in Progress, April 2004.

10. Authors' Addresses

Patrik Faltstrom
Cisco Systems Inc
Ledasa
273 71 Lovestad
Sweden

EMail: paf@cisco.com
URI: http://www.cisco.com

Michael Mealling
VeriSign
21345 Ridgetop Circle
Sterling, VA 20166
US

Email: michael@verisignlabs.com
URI: http://www.verisignlabs.com

11. Full Copyright Statement

Intellectual Property

Acknowledgement

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