

Network Working Group
Internet-Draft
Intended status: Informational
Expires: March 3, 2011

B. Carpenter
Univ. of Auckland
S. Jiang
Huawei Technologies Co., Ltd
B. Zhou
ChinaMobile
August 30, 2010

Problem Statement for Referral
draft-carpenter-referral-ps-01

Abstract

The purpose of a referral is to enable a given entity in a multiparty Internet application to pass information to another party. It enables a communication initiator to be aware of relevant information of its destination entity before launching the communication. This memo discusses the problems involved in referral scenarios.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 3, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Terminology	3
3. Goals of Referral	4
3.1. Reachability	4
3.2. Path Selection	4
3.2.1. An Example: Triangle Path Optimization	4
3.3. Interface Selection	5
4. Problem Statement	6
4.1. IP Addresses are not sufficient	6
4.2. FQDNs are not sufficient	7
4.3. Relevant Information is lack	8
4.4. Extra complexity from ID-Locator Split Mechanisms	9
5. A Generic Referral Mechanism is needed	9
6. Security Considerations	11
7. IANA Considerations	11
8. Acknowledgements	11
9. Change log	12
10. References	12
10.1. Normative References	12
10.2. Informative References	12
Authors' Addresses	13

1. Introduction

A frequently occurring situation is that one entity A connected to the Internet (or to some private network using the Internet protocol suite) needs to be aware of the information of another entity B in order to reach it. The information can be obtained from B itself or some third-party entity C. This is known as a referral.

Referral is the act whereby one entity informs another entity how to contact a specific entity. It enables a communication initiator to be aware of relevant information of its destination entity in order to launch a communication channel. This referral information can be obtained through an existing communication channel between these two entities or from third-party entities.

In the original design of the Internet, IP addresses were global, unique, and quasi-permanent. Also any differentiation beyond that provided by an IP address was done by protocol and port numbers. Referrals were therefore performed simply by passing an IP address and possibly protocol and port numbers. In fact simple referrals (the first case above, sometimes called first-party referrals) were never needed since A could simply use B's address. Third-party referrals were trivial: C would tell A about B's address. Thus, it became common practice to pass raw addresses between entities. A classical example is the FTP PORT command [RFC0959].

2. Terminology

This document makes use of the following terms:

- o "Entity": we use this rather than "application" to describe any software component embedded in an Internet host, not just a specific application, that sends, receives or makes use of referrals. Also, in case of dynamic load sharing or failover, an entity might even migrate between hosts.
- o "Referral": the act of one entity informing another entity how to contact a specific entity.
- o "Reference": the actual data (name, address, identifier, locator, pointer, etc.) that is the basis of a referral.
- o "Referring entity": the entity that sends a referral.
- o "Receiving entity": the entity that receives a referral.
- o "Referenced entity": the target entity of a reference.
- o "Scope": the region or regions of the Internet within which a given reference is applicable to reach the referenced entity.

3. Goals of Referral

The principal purpose of referral is to enable one entity in a multi-party application to pass information to another party involved in the same application. This document makes no assumptions about whether the entities are acting as clients, servers, peers, super-nodes, relays, proxies, etc., as far as the application is concerned. Neither does it take a position as to how the various entities become aware of the need to send a referral; this depends entirely on the structure of the application.

3.1. Reachability

The primary goals of referral is to enable a communication initiator to reach its destination entity. Referral is a best effort mechanism. It does not guarantee actual reachability, since the referring entity has no general way of knowing which paths exist between the receiving entity and the referenced entity. Even if a reference is theoretically in scope, and within its defined lifetime, it may have become unreachable since it was sent. A receiving entity should always be prepared for reachability failures and associated retry and failover mechanisms, which are out of scope for the referral mechanism itself.

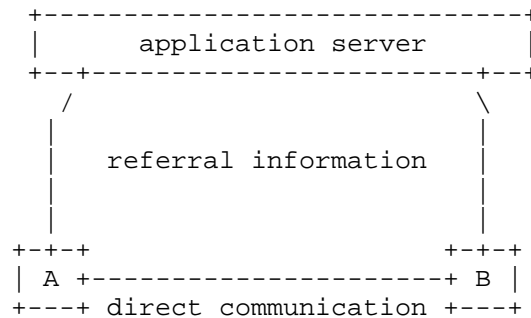
3.2. Path Selection

A reference might carry multiple references for the same target. These may lead to multiple possible paths from the receiving entity to the referenced entity. This scenario is particularly generic when the destination or/and source entity has multiple interfaces or is multi-homed.

The referring entity is not likely to know which path is best. The receiving entity will need to make a choice, possibly by local policy (e.g. [RFC3484]) or possibly by trial and error (e.g. [RFC4038], [RFC5534]). This choice is also out of scope for the referral mechanism itself.

3.2.1. An Example: Triangle Path Optimization

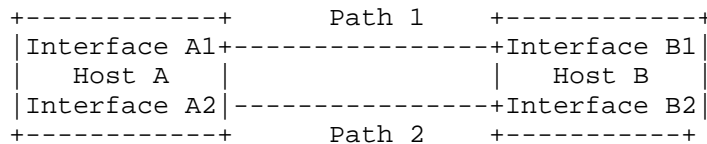
In application scenarios, the triangular path shown below is common. Both Host A and Host B connect to an application server and the application server forwards traffic as a relay agent. A slightly more complicated scenario is when the two hosts connect to different application servers individually and application servers talk to each other's relay agents. In SIP, this is often called the "SIP trapezoid".



By passing A's reference to B, B can try to communicate directly with A, using the communication line at the bottom. If the direct communication is established successfully, the triangle path gets optimized. Both the application server and network bandwidth can be benefit from this operation.

3.3. Interface Selection

We also encounter multi-interfaced hosts whose reachability is bound to a particular (logical/physical) interface. More information is required to indicate which interface may be used under different circumstances. Multi-interface is defined and studied by IETF MIF WG. Here referral can provide the host A's multi-interface information to host B, accordingly, host B can select one of the interface to establish the connection.



For example, as shown in the above figure, Host A has connected to Host B through Path 1. They can exchange references through Path 1. They may find out Path 2 using different interfaces is better than Path 1, maybe cheaper, faster or more stable. Then, they can switch to Path 2. Host A has interface A1 as broadband access, almost free; and interface A2 is 3G access, which costs 0.1 \$ per MB. Both of them are available for incoming connection. If these information is passed to host B, through referral, then host B should choose A1 interface to reach host A. This kind of information is useful to express the source's status or preference.

In order to choose between different interfaces, not only the connectivity information of these interfaces, but also some additional

information may be helpful, such as bandwidth, finance cost, latency, etc. These additional information may also be provided through referral. However, this additional information, even if known by the referring entity, may be invalid at the location of the receiving entity.

4. Problem Statement

Unfortunately, the simple approach to referrals, passing an IP address, often fails in today's Internet. As has been known for some time [RFC2101], hosts' IP addresses no longer all have global scope. They often have limited reachability, and may have limited lifetime. They are not sufficient to establish communication in many cases of dynamic referrals, for a variety of reasons. FQDN may be used instead in some scenarios. However, FQDN also has its own limitation and may fail in some scenarios.

4.1. IP Addresses are not sufficient

It is no longer reasonable to assume that a host with a fixed location has a fixed IP address, or even a stable IP address.

Furthermore, in the context of IPv4 address exhaustion, several solutions have emerged to share a single public IPv4 address between several customers simultaneously. Consequently, a public IPv4 address often no longer identifies a single customer/user/host while a private IPv4 address is meaningless out of the private network scope. Other information (e.g., port range) is required to identify unambiguously a given customer/user/host. Both IP addresses and port numbers may be different on either side of a NAT or some other middlebox [RFC3234], and firewalls may block them. It is no longer reasonable to assume that an IP address for a host, which allows a given peer to reach that host in one location, also works from a different location - even if that host is reachable from the second location.

Also, the Internet now has two co-existing address formats for IPv4 and IPv6. Direct communication can only be established when both peers use the same IP version. Having the address of the source and destination in the same IP version does not necessarily mean that the path will be using that IP version. Simple approaches may cause unnecessary double translation [I-D.boucadair-softwire-cgn-bypass]. Some addresses may even be the result of translation between IPv4 and IPv6, with severe limitations on their scope and lifetime. Sending an out-of-scope or expired address, or one of the wrong format, as a referral will fail.

IP addresses today may have an implied "context" (VPN, VoIP VC, IP TV, etc.): the reachability of such an address depends on that context.

An implication of these issues is that there is no clean definition of the scope of an address (especially an IPv4 address, due to the prevalence of NAT). It is impossible to determine algorithmically, by inspecting the bits of an address, what its scope of reachability is. Resolving this problem would greatly clarify the general problem of referrals.

4.2. FQDNs are not sufficient

In some cases, this problem may be readily solved by passing a Fully Qualified Domain Name (FQDN) instead of an IP address. Indeed, that is an architecturally preferred solution [RFC1958]. However, it is not sufficient in many cases of dynamic referrals. Experience shows that an application cannot use a domain name in order to reliably find usable address(es) of an arbitrary peer. Domain names work fairly well to find the addresses of public servers, as in web servers or SMTP servers, because operators of such servers take pains to make sure that their domain names work. But DNS records are not as reliably maintained for arbitrary hosts such as might need to be contacted in peer-to-peer applications, or for servers within corporate networks. Many small networks do not even maintain DNS entries for their hosts, and for some networks that do list local hosts in DNS, the listings may well be unusable from a remote location, because of two-faced DNS, or because the A record contains a private address. These cases may even be intentional as part of a security ring-fence, where w3.example.com only resolves within the corporate boundary, and/or resolves to IP addresses which are only reachable within the corporate administrative boundaries. In such contexts, incoming connections are usually filtered by the corporate firewall.

An additional issue with FQDNs is the very common situation where multiple hosts are hidden behind a NAT, but they share one FQDN which is in fact a dummy name, created automatically by the ISP so that reverse DNS lookup will succeed for the NAT's public IPv4 address. Such FQDNs are useless for identifying specific hosts.

Furthermore, an FQDN may not be sufficient to establish successful communications involving heterogeneous peers (i.e., IPv4 and IPv6) since A and AAAA records may not be consistently provisioned. There are known cases where a server has one name that produces an A record (e.g., www.example.com) and another name that produces an AAAA record (e.g., ipv6.example.com). An additional complication is that some answers from DNS may be synthetic IP addresses, e.g., AAAA records

sent by DNS64. The host may have no means to detect that such an address represents an IPv4 host. These addresses should not be interpreted as native IPv6 address.

In such cases, an IP address either cannot be derived from an FQDN, or if so derived, cannot be accessed from an arbitrary location in the Internet.

A related problem is that an application does not have a reliable way of knowing its own domain name - or to be more precise, a way of knowing a domain name that will allow the application to be reached from another location.

There are wider systemic problems with the DNS as a reliable way to find a usable address, which are somewhat out of scope here, but can be summarised:

- o In large networks, it is now quite common that the DNS administrator is out of touch with the applications user or administrator, and as a result, that the DNS is out of sync with reality.
- o DNS was never designed to accommodate mobile or roaming hosts, whose locator may change rapidly.
- o DNS has never been satisfactorily adapted to isolated, transiently-connected, or ad hoc networks.
- o It is no longer reasonable to assume that all addresses associated with a DNS name are bound to a single host. One result is that the DNS name might suffice for an initial connection, but a specific address is needed to rebind to the same peer, say, to recover from a broken connection.
- o It is no longer reasonable to assume that a DNS query will return all usable addresses for a host.
- o Hosts may be identified by a different URI per service: no unique URI scheme, meaning no single FQDN, will apply.

For all the above reasons, the problem of address referrals cannot be solved simply by recommending the use of FQDNs instead. The guideline in [RFC1958] is in fact too simple for today's network. Something more elaborate than an IP address or an FQDN appears to be needed in the general case of application referrals.

4.3. Relevant Information is lack

Neither an IP address nor an FQDN gives complete information about the referenced entity. For example, IP addresses normally have associated lifetimes (derived from DHCP, SLAAC or the relevant DNS TTL), so they should be treated as invalid after their lifetimes expire. A referral that does not convey the lifetime associated with an address is problematic. As mentioned above, the scope of a

reference also affects its usefulness. These are examples of additional information that is necessary to correctly interpret a referral; therefore part of the problem is conveying such information along with the reference.

4.4. Extra complexity from ID-Locator Split Mechanisms

Additional complexity for referrals would come from the deployment of any technology that separates locators from identifiers, rather than combining the two as an IP address. Since a very wide range of such solutions have been proposed (e.g. HIP, LISP, ILNP and Name-based Sockets) [I-D.ubillos-name-based-sockets], it is difficult to define the resulting problems precisely.

However, to consider the example of Name-based Sockets, if a referral was made based on the IP address being used at a given instant for a Name-based Socket, that address might be useless by the time the referral was completed, because the socket suddenly migrated to a different IP address.

The SHIM6 protocol [RFC5533] and the Multiple Interfaces (mif) Working Group may produce similar difficulties, since they also consider scenarios where the IP address in use for some purpose may change unexpectedly.

Any referral mechanism must be able to deal with situations where the locator corresponding to a given identifier is subject to change.

5. A Generic Referral Mechanism is needed

The first motivation is the observation that unless the parties involved have reached an understanding about the scope, lifetime, and format of the elements in a referral through some other means, that information must be passed with the referral. This is required so that the receiving entity can determine whether or not the referral is useful. The referral therefore needs to consist of a fully-fledged data structure, or to be made using a mutually agreed referral protocol.

When an attempt to establish a communication channel based on certain referral information fails, good design suggests that the receiving entity should attempt to correct the situation. For example, if communication fails to be established using an IP address, it would often be appropriate to attempt a DNS lookup, despite the difficulties mentioned above. The second motivating problem is that it may be helpful to the entity receiving a reference to also receive information about the source of the reference, such as an FQDN, if

that is known to the sender of the reference. The receiving entity can then attempt to recover a valid address (and possibly port number) for the referred entity.

The third motivating problem is to allow a reference to contain alternatives to an IP address or an FQDN, when any such alternatives exist.

Additional arguments for a generic referral mechanism include:

1. Allow for general mechanisms that can be used by any application to handle references and understand the meaning of referral information, such as IP address, possibly protocol and port numbers. However, there is an open question whether this standard referral design should be used for new applications only, or extended to existing applications.
2. Simplify ALG design during middlebox traversal. There are middleboxes, like firewalls and translators, especially in the mobile network, which require application layer gateways ALG. The cost of ALG functions is huge for the mobile operator in terms of implementation, performance. Standard references could simplify ALG implementation during middlebox traversal in the mobile network.
3. Simplify packet inspection. Operators sometimes need to inspect information or details during communication for administration reasons. If referral mechanism is standardized, it is easier for an operator to capture and investigate the required information.

We observe that we have identified two general requirements: the need to define address scope more precisely, and the need to communicate references in a generic way.

It should be noted that partial or application-specific solutions to these problems abound, because any multi-party distributed application must solve them. The best documented example is ICE [RFC5245], which is an active protocol specific to applications mediated by SDP [RFC4566]. ICE "works by including a multiplicity of IP addresses and ports in SDP offers and answers, which are then tested for connectivity by peer-to-peer connectivity checks." The question raised here is whether we can define requirements for a generic solution that can be used by future applications, and possibly be retro-fitted to existing applications.

One approach could be a "SuperICE" designed to be completely general and not tied to the SDP model. Another approach is the idea of a generic referral object. Such an object could be passed between the entities of a multi-party application, but without defining a specific protocol for that purpose. Some applications might choose to send it in-band as a raw binary object, others might use a simple

ASCII encoding, and still others might prefer to encode it in XML, for example. Finally, it might also be used as part of SuperICE.

6. Security Considerations

It should be noted that referral should not function as a way to nullify the effect of a firewall or any other security mechanism. If the receiving entity chooses a particular reference and attempts to send packets to the corresponding IP address, whether they are delivered or not will depend on the existing security mechanisms, whatever they may be.

Nevertheless, if a site security policy requires it, certain references may be excluded from referral information sent to certain destinations. This would require a security policy mechanism to be added to the process of generating referral information.

Forged or intercepted referral information would enable a wide variety of attacks. Although not fundamentally different from attacks based on forged or observed IP addresses or FQDNs, no doubt referral would allow such attacks to be more ingenious, simply because they provide more information than an address or FQDN alone. Referral information should be transmitted through authenticated and encrypted channels. It is not further elaborated here.

Referral may raise potential privacy issues, which are not explored in this document. For example, in the SIP context, mechanisms such as [RFC3323] and [RFC5767] are available to hide information that might identify end-points. Referral usage scenarios must ensure that they do not unintentionally defeat privacy solutions.

7. IANA Considerations

This document requests no action by IANA.

8. Acknowledgements

Valuable comments and contributions were made by Mohamed Boucadair, Dan Wing, Keith Moore and others.

This document was produced using the xml2rfc tool [RFC2629].

9. Change log

draft-carpenter-referral-ps-00: original version, 2010-06-21.

draft-carpenter-referral-ps-01: add content regarding to ID-Locator Split Mechanisms, 2010-08-30.

10. References

10.1. Normative References

- [RFC0959] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, October 1985.
- [RFC3323] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, November 2002.
- [RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.
- [RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, June 2009.
- [RFC5534] Arkko, J. and I. van Beijnum, "Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming", RFC 5534, June 2009.

10.2. Informative References

- [I-D.boucadair-softwire-cgn-bypass]
Boucadair, M., "Procedure to bypass DS-lite AFTR (work in progress)", December 2009.
- [I-D.ubillos-name-based-sockets]
Ubillos, J., "Name Based Sockets (work in progress)", July 2010.
- [RFC1958] Carpenter, B., "Architectural Principles of the Internet", RFC 1958, June 1996.

- [RFC2101] Carpenter, B., Crowcroft, J., and Y. Rekhter, "IPv4 Address Behaviour Today", RFC 2101, February 1997.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [RFC2775] Carpenter, B., "Internet Transparency", RFC 2775, February 2000.
- [RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, February 2002.
- [RFC4038] Shin, M-K., Hong, Y-G., Hagino, J., Savola, P., and E. Castro, "Application Aspects of IPv6 Transition", RFC 4038, March 2005.
- [RFC5767] Munakata, M., Schubert, S., and T. Ohba, "User-Agent-Driven Privacy Mechanism for SIP", RFC 5767, April 2010.

Authors' Addresses

Brian Carpenter
Department of Computer Science
University of Auckland
PB 92019
Auckland, 1142
New Zealand

Email: brian.e.carpenter@gmail.com

Sheng Jiang
Huawei Technologies Co., Ltd
Huawei Building, No.3 Xinxu Rd.,
Shang-Di Information Industry Base, Hai-Dian District, Beijing
P.R. China

Email: shengjiang@huawei.com

Bo Zhou
China Mobile
Unit2, 28 Xuanwumenxi Ave,Xuanwu District
Beijing, 100053
P.R. China

Email: zhouboyj@gmail.com

Internet Engineering Task Force
Internet-Draft
Intended status: BCP
Expires: April 21, 2011

A. Durand
Juniper Networks
I. Gashinsky
Yahoo! Inc.
D. Lee
Facebook, Inc.
S. Sheppard
ATT Labs
October 18, 2010

Logging recommendations for Internet facing servers
draft-durand-server-logging-recommendations-00

Abstract

In the wake of IPv4 exhaustion and deployment of IP address sharing techniques, this document recommends that Internet facing servers log port number and accurate timestamps in addition to the incoming IP address.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 21, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Recommendations	3
3. ISP Considerations	4
4. IANA Considerations	4
5. Security Considerations	4
6. References	5
6.1. Normative references	5
6.2. Informative references	5
Authors' Addresses	5

1. Introduction

According to the most recent predictions, the global IPv4 address free pool at IANA will exhaust sometime in 2011. After that, service providers will have a hard time finding enough IPv4 global addresses to sustain product and subscriber growth. Due to the huge global existing infrastructure, both hardware and software, vendors and service providers must continue to support IPv4 technologies for the foreseeable future. As legacy applications and hardware are retired the reliance on IPv4 will diminish but this is a years long perhaps decades long process.

To maintain legacy IPv4 address support, service providers will have little choice but to share IPv4 global addresses among multiple customers. Techniques to do so are outside of the scope of this documents. All include some form of address translation/address sharing, being NAT44, NAT64 or DS-Lite.

The effects on the Internet of the introduction of those address sharing techniques have been documented in [I-D.ietf-intarea-shared-addressing-issues].

Address sharing techniques come with their own logging infrastructure to track the relation between which original IP address and source port(s) were associated with which user and external IPv4 address at any given point in time. In the past to support abuse mitigation or public safety requests, the knowledge of the external global IP address was enough to identify a subscriber of interest. With address sharing technologies, only providing information about the external public address associated with a session to a service provider is no longer sufficient information to unambiguously identify customers.

Note: this document provides recommendations for Internet facing servers logging incoming connections. Its does not provide any recommendations about logging on carrier-grade NAT or other address sharing tools.

2. Recommendations

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

It is RECOMMENDED as best current practice that Internet facing servers logging incoming IP addresses also log:

- o The source port number.
- o A timestamp accurate to the second, with associated time zone.
- o The transport protocol (usually TCP or UDP) and destination port number, when the server application is defined to use multiple transports or multiple ports.

Discussion: Carrier-grade NATs may have different policies to recycle ports, some implementations may decide to reuse ports almost immediately, some may wait several minutes before marking the port ready for reuse. As a result, servers have no idea how fast the ports will be reused and, thus, should log timestamps using a reasonably accurate clock. At this point the RECOMMENDED accuracy for timestamps is to the second or better.

Examples of Internet facing servers include, but are not limited to, web servers and email servers.

Although the deployment of address sharing techniques is not immediately foreseen in IPv6, the above recommendations apply to both IPv4 and IPv6, if only for consistency and code simplification reasons.

Discussions about data retention policies are out of scope for this document.

3. ISP Considerations

ISP deploying IP address sharing techniques should also deploy a corresponding logging architecture to maintain records of the relation between customers identity and IP/port resources they utilize. However, recommendation on this topic are out of scope for this document.

4. IANA Considerations

None.

5. Security Considerations

In the absence of source port number and accurate timestamp, operators deploying any address sharing techniques will not be able to identify unambiguously customers when dealing with abuse or public safety queries.

6. References

6.1. Normative references

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

6.2. Informative references

[I-D.ietf-intarea-shared-addressing-issues]
Ford, M., Boucadair, M., Durand, A., Levis, P., and P.
Roberts, "Issues with IP Address Sharing",
draft-ietf-intarea-shared-addressing-issues-02 (work in
progress), October 2010.

Authors' Addresses

Alain Durand
Juniper Networks
1194 North Mathilda Avenue
Sunnyvale, CA 94089-1206
USA

Email: adurand@juniper.net

Igor Gashinsky
Yahoo! Inc.
45 West 18th St.
New York, NY 10011
USA

Email: igor@yahoo-inc.com

Donn Lee
Facebook, Inc.
1601 S. California Ave.
Palo Alto, CA 94304
USA

Email: donn@facebook.com

Scott Sheppard
ATT Labs
575 Morosgo Ave, 4d57
Atlanta, GA 30324
USA

Email: Scott.Sheppard@att.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 11, 2011

P. Faltstrom, Ed.
Cisco
P. Hoffman, Ed.
VPN Consortium
June 9, 2011

The Unicode code points and IDNA - Unicode 6.0
draft-faltstrom-5892bis-05.txt

Abstract

This memo documents IETF consensus for IDNA derived character properties related to the three code points, existing in Unicode 5.2, that changed property values when version 6.0 was released. The consensus is that no update is needed to RFC 5892 based on the changes made in Unicode 6.0.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 11, 2011.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as

described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. U+0CF1 KANNADA SIGN JIHVAMULIYA	3
1.2. U+0CF2 KANNADA SIGN UPADHMANIYA	3
1.3. U+19DA NEW TAI LUE THAM DIGIT ONE	3
2. IETF Consensus	3
3. IANA Considerations	3
4. Security Considerations	4
5. Acknowledgements	4
6. Normative References	4
Authors' Addresses	4

1. Introduction

RFC 5892 [RFC5892] specifies an algorithm that was defined when version 5.0 (later updated to version 5.2) [Unicode5.2] was the current version of Unicode, and it also defines a derived property value based on that algorithm. Unicode 6.0 [Unicode6] has changed GeneralCategory of three code points that were allocated in Unicode 5.2 or earlier. This implies the derived property value differs depending on whether the property definitions used are from Unicode 5.2 or 6.0. These are non-backward-compatible changes as described in section 5.1 of RFC 5892.

The three code points are:

1.1. U+0CF1 KANNADA SIGN JIHVAMULIYA

The GeneralCategory for this character changes from So to Lo. This implies that the derived property value changes from DISALLOWED to PVALID.

1.2. U+0CF2 KANNADA SIGN UPADHMANIYA

The GeneralCategory for this character changes from So to Lo. This implies that the derived property value changes from DISALLOWED to PVALID.

1.3. U+19DA NEW TAI LUE THAM DIGIT ONE

The GeneralCategory for this character changes from Nd to No. This implies that the derived property value changes from PVALID to DISALLOWED.

2. IETF Consensus

No change to RFC 5892 is needed based on the changes made in Unicode 6.0.

This consensus does not imply that no changes will be made to RFC 5892 for all future updates of The Unicode Standard.

This RFC is being produced because 6.0 is the first version of Unicode to be released since IDNA2008 was published.

3. IANA Considerations

IANA is to update the derived property value registry according to

RFC 5892 and property values as defined in The Unicode Standard version 6.0.

4. Security Considerations

When the algorithm presented in RFC 5892 is applied using the property definitions of Unicode Standard Version 6.0, the result will be different from when it is applied using the property definitions of Unicode 5.2 for the three code points discussed in this document in addition to the changes for code points being unassigned in Unicode 5.2. The three code points are unlikely to occur in internationalized domain names, however, so the security implications of the changes are minor.

5. Acknowledgements

The main contributors are (in alphabetical order) Eric Brunner-Williams, Vint Cerf, Tina Dam, Martin Duerst, John Klensin, Mark Davis, Pete Resnick, Markus Scherer, Andrew Sullivan, Kenneth Whistler and Nicholas Williams.

Not all contributors believe the solution for the issues discussed in this document is optimal.

6. Normative References

[RFC5892] Faltstrom, P., "The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)", RFC 5892, August 2010.

[Unicode5.2]
The Unicode Consortium, "The Unicode Standard, Version 5.2.0", Unicode 5.0.0, Boston, MA, Addison-Wesley ISBN 0-321-48091-0, as amended by Unicode 5.2.0
<http://www.unicode.org/versions/Unicode5.2.0/>, 2009,
<<http://www.unicode.org/versions/Unicode5.2.0/>>.

[Unicode6]
The Unicode Consortium, "The Unicode Standard, Version 6.0.0", October 2010.

Authors' Addresses

Patrik Faltstrom (editor)
Cisco

Email: paf@cisco.com

Paul Hoffman (editor)
VPN Consortium

Email: paul.hoffman@vpnc.org

Network Working Group
Internet-Draft
Intended status: BCP
Expires: April 21, 2011

E. Lear
Cisco Systems GmbH
P. Eggert
UCLA
October 18, 2010

IANA Procedures for Maintaining the Timezone Database
draft-lear-iana-timezone-database-00

Abstract

ATTENTION: This memo contains a DRAFT proposal for the IANA to assume operational responsibilities relating to the management of the Timezone (TZ) Database. The authors seek comment and review of this proposal. No action will be taken without rough consensus of the TZ community.

The Timezone (TZ) Database consists of timezone information for all localities throughout the world. This database has been meticulously maintained and distributed free of charge by a group of volunteers, coordinated by a single volunteer who is now planning to retire. This memo specifies a DRAFT PROPOSAL for the IANA procedures involved with maintenance of the TZ database and associated code, including how to submit proposed updates, how decisions for inclusion of those updates are made, and the selection of a designated expert BY AND FOR the timezone community. The intent of this memo is, to the extent possible, document existing practice and provide a means to ease succession.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>.

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

This Internet-Draft will expire on April 21, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the BSD License.

1. Introduction

ATTENTION: This memo contains a DRAFT proposal for the IANA to assume operational responsibilities relating to the management of the Timezone (TZ) Database. The authors seek comment and review of this proposal. No action will be taken without rough consensus of the TZ community.

Since the early 1980s, a database that is in use on nearly all UNIX systems, Java systems, and other sorts of systems has been hosted at the National Institutes of Health. [TZDB] The database consists of both historic and current entries for geographies throughout the world. Associated with the database is a reference implementation of functions that can be used to convert time values.

The database has been maintained by volunteers that participate in a mailing list that is also hosted at the NIH. The database itself is updated approximately twenty times per year, depending on the year, based on information these experts provide to the maintainer. Arthur David Olson has maintained the database, coordinated the mailing list, and provided a release platform since the database's inception. With his retirement now approaching it is necessary to provide a means for this good work to continue. The Internet community owes Arthur Olson and the volunteers on the tz mailing list a debt of gratitude.

The IANA provides registry services to the Internet community. Those

registries are coordinated by technical experts who are designated by the Internet Engineering Steering Group (IESG). The IANA is also well suited as a distribution platform for the TZ database itself.

The IETF has for quite some time had the capability to maintain non-working group mailing lists. The TZ mailing list would fit nicely just as such a list.

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 RFC 2119 [RFC2119].

TZ Database The TimeZone Database, sometimes referred to as the Olson Database. This database consists of information about offsets from UTC for different localities, including daylight savings time (DST) transition information.

TZ Coordinator The person or people who maintain and manage release of the TZ Database. The TZ coordinator also has responsibility for maintaining the TZ mailing list. The TZ coordinator is a Designated Expert, as defined in [RFC5226].

TZ mailing list The forum where matters relating to the TZ database and supporting code are discussed.

The rest of this document specifies the following:

1. Transferring and maintenance of the TZ mailing list;
2. Procedures for selecting a technical expert for the technical expert who will play the role of coordinator, as well as release manager for the TZ database;
3. Procedures for updating the TZ database;
4. Maintenance and ownership of reference code; and
5. Ownership of the database.

2. The TZ Mailing List

For many years the TZ mailing list at the NIH has been the forum where discussion of changes to the TZ database and support files would take place. In addition, the TZ mailing list is used to announce releases of the database. Currently the TZ mailing list is administered by the TZ coordinator.

This list membership will be transitioned to the IETF mail server. The TZ coordinator will continue to manage the list, in accordance with rules of governance for non-WG mailing lists (including, for example, the commonly used "Note Well" statement). The list will be

used just as it has been, to learn of, discuss, and confirm TZ definition changes, as well as an announcement list for new versions of the database. The TZ coordinator will continue to manage the list.

3. Making Updates to the TZ Database

Updates to the TZ database are made by the TZ coordinator in consultation with the TZ mailing list. TZ coordinator is empowered to decide, as the designated expert, appropriate changes, but SHOULD take into account views expressed on the mailing list.

The TZ coordinator will also decide the timing of database releases. The release itself today consists of several tar files that are downloaded from a well known location.

Moving forward, the TZ database is to be signed prior to release using a well known key, along with any appropriate supporting information and distributed from a well known location that is advertised by IANA in a manner of its choosing.

4. Selecting or Replacing a TZ Coordinator

From time to time it will be necessary to replace a TZ Coordinator. This could occur for a number of reasons:

- o The coordinator is retiring (as Arthur Olson is) or has announced that he or she will be unable to continue to perform the function;
- o The coordinator is missing or has died;
- o The coordinator is not performing the function in accordance with community wishes.

In any of these cases, members of the community should raise the issue on the TZ list. If a rough consensus can be formed easily, and quickly, then the results should be presented to the IESG for comment and review. In keeping with [RFC5226], the IESG selects the TZ coordinator(s). The IESG MUST use rough consensus of the TZ mailing list as their primary guide to further action, when it exists. If the IESG determines that there is no rough consensus within the TZ community, the IESG will assign one of its members to develop that rough consensus on the TZ mailing list, and through whatever other means may be necessary. If rough consensus still cannot be developed after one month, at the discretion of the IESG, it MAY then choose a replacement TZ coordinator. The IESG is not an avenue for appeals of specific decisions of the coordinator, but rather a last resort when a coordinator is thought not to be functioning in an appropriate way.

N.B., the coordinator is a function, and may be filled by one OR MORE people, as the community sees fit.

5. Maintenance and Distribution of Reference Code

Currently the maintainer of the TZ database also maintains reference code. This software is currently distributed under the BSD license. No change shall be made to the license without consultation and rough consensus of the community. IANA shall allow for the downloading of this reference code. The reference implementation shall be distributed along with an associated cryptographic signature of an identity that IANA shall publish.

6. Database Ownership

It is the understanding of the IESG, ISOC, and IANA that the database itself is public domain. Certain portions of code currently distributed fall under the BSD license, and will be distributed as such. Should claims be made and substantiated against the database, the IANA will act in accordance with all competent court orders. No further ownership claims will be made by IANA, the IETF Trust, or ISOC on the database.

7. IANA Considerations

The IANA will see that the role of TZ Coordinator is filled, based on the procedures described above. The IANA will act as a repository for the TZ database and associated reference code. The database coordinator will be named by the IESG as described above, and will act as the maintainer of the database and code, as described above. The IANA will provide the TZ coordinator with appropriate access to maintain the database, as well as necessary tooling that may be required, so long as no direct software costs are incurred. Both current and historical versions of the database will be stored and distributed via HTTP/HTTPS. IANA will be operationally responsible for the security of the system upon which the database resides.

The IANA will also maintain a cryptographic identity that is used to sign the database, and that will survive a change of coordinators.

8. Security Considerations

The distribution of the database is currently not secured. This memo states that moving forward the TZ database will be distributed with a

valid cryptographic signature.

9. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [TZDB] Eggert, P. and A. Olson, "Sources for Time Zone and Daylight Saving Time Data",
<<http://www.twinsun.com/tz/tz-link.htm>>.

Appendix A. Changes

- o Initial Revision

Authors' Addresses

Eliot Lear
Cisco Systems GmbH
Richtistrasse 7
Wallisellen, ZH CH-8304
Switzerland

Phone: +41 1 878 9200
Email: lear@cisco.com

Paul Eggert
UCLA
Computer Science Department
4532J Boelter Hall
Los Angeles, CA 90095
USA

Phone: +1 310 267 2254
Email: eggert@cs.ucla.edu

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 28, 2011

L. Masinter
Adobe
October 25, 2010

MIME and the Web
draft-masinter-mime-web-info-01

Abstract

This document describes some of the ways in which parts of the MIME system, originally designed for electronic mail, have been used in the Web, and some of the ways in which those uses have resulted in difficulties. Given this background and justification, this document then goes on to outline requirements for changes to MIME registries and practices for their use within W3C and IETF, in order to address those difficulties. Within IETF, a companion Best Current Practice document will be developed to specifically make some changes to the Internet Media Types and Charset registries.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 28, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. History	3
2.1. Origins of MIME	3
2.2. Introducing MIME into the Web	4
2.3. Distributed Extensibility	5
3. Problems with application to the Web	5
3.1. Lack of clarity	5
3.2. Differences between email and Web delivery	6
3.3. The Rules Weren't Quite Followed	7
3.4. Consequences	7
3.5. The Down Side of Extensibility	8
4. Additional considerations	8
4.1. There are related problems with charsets	8
4.2. Embedded, downloaded, launch independent application	9
4.3. Additional Use Cases: Polyglot and Multiview	9
4.4. Evolution, Versioning, Forking	9
4.5. Content Negotiation	10
4.6. Fragment identifiers	11
5. Recommendations	11
5.1. Internet Media Type registration	12
5.1.1. MIME registry magic numbers for sniffing	12
5.1.2. Scripting and scriptable content safety	12
5.1.3. Fragment identifiers	12
5.1.4. Application info	12
5.1.5. File extensions in registry	12
5.2. Sniffing	13
5.2.1. Sniffing uses Media Type magic number	13
5.2.2. Sniffing when there are multiple different definitions	13
5.2.3. Sniffing charsets	13
5.2.4. Sniffing security uses scriptability info	13
5.3. Changes to IANA processes for MIME registries	13
5.4. FTP specification	13
5.5. Update some URI definitions	14
5.6. Changes to W3C findings, processes	14
6. Acknowledgements	14
7. IANA Considerations	14
8. Security Considerations	14
9. Informative References	14
Author's Address	15

1. Introduction

This document was initially prompted by a set of discussions about Web architecture and the difficulties surrounding evolution of the Web, Internet Media types, multiple specifications for a single media type, and related discussions.

The document gives some of the history of MIME and its introduction and use in the web Section 2. It then describes some of the current difficulties with the use of MIME in the web context Section 3. This background and context is then followed by a description of changes which would reduce some of those difficulties; the changes involve specifications, practices, and registries within IETF and W3C Section 5. In particular, changes to the registry and maintenance procedures for MIME-related registries maintained by IANA are describes.

Currently, discussion of this document is suggested on the mailing list www-tag@w3c.org (mailing list open for subscription to all), archives at <http://lists.w3.org/Archives/Public/www-tag/>.

NOTE: This document is still quite rough; some of the facts need to be checked, many sections still need expansion. Any help with references and such appreciated.

2. History

2.1. Origins of MIME

MIME ("Multipurpose Internet Mail Extensions") was invented originally for email, based on general principles of "messaging" (a foundational architecture framework). The role of MIME was to extend Internet email messaging from ASCII-only plain text, to include other character sets, images, rich documents, etc.) [RFC1521], [RFC1522]. The basic architecture of complex content messaging is:

- o Message sent from A to B.
- o Message includes some data. Sender A includes standard 'headers' telling recipient B enough information that recipient B knows how sender A intends the message to be interpreted.
- o Recipient B gets the message, interprets the headers for the data and uses it as information on how to interpret the data.

MIME is a "tagging and bagging" specification:

tagging: How to label content so the intent of how the content should be interpreted is known.

bagging: How to wrap the content so the label is clear, or, if there are multiple parts to a single message, how to combine them.

"MIME types" (renamed "Internet Media Types" in later specs [RFC2046]) are part of the "tagging" -- a way to describe the content of a message so that it could be used to initiate interpretation of a message. The "Internet Media Type registry" (MIME type registry) is where someone can tell the world what a particular label means, as far as the sender's intent of how recipients should process a message of that type, and the description of a recipients capability and ability for senders.

2.2. Introducing MIME into the Web

The original World Wide Web (the 0.9 version of HTTP, see [RFC1945]) didn't have "tagging and bagging" -- everything sent via HTTP was assumed to be HTML. However, at the time (early 1990's) other distributed information access systems, including Gopher (distributed menu system) and WAIS (remote access to document databases) were adding capabilities for accessing many things other text and hypertext and the WWW folks were considering type tagging. It was agreed that HTTP should use MIME as the vocabulary for talking about file types and character sets. The result was that HTTP 1.0 added the "content-type" header, following (more or less) MIME. Later, for content negotiation, additional uses of this technology (in 'Accept' headers) were also added.

The differences between the use of Internet Media Types between email and HTTP have minor:

- o default charset: HTTP specified ISO-8859-1 as the default character set, not US-ASCII
- o requirement for CRLF in plain text: in practice, web clients didn't restrict content to use CRLF in text/* MIME bodies.

These minor differences have caused a lot of trouble.

2.3. Distributed Extensibility

The real advantage of using Internet Media Types to label content meant that the Web was no longer restricted to a single format. This one addition meant expanding from Global Hypertext to Global Hypermedia (as suggested in a 1992 email [connolly92])

```
+-----+
| The Internet currently serves as the backbone for a global |
| hypertext.  FTP and email provided a good start, and the gopher, |
| WWW, or WAIS clients and servers make wide area information |
| browsing simple.  These systems even interoperate, with email |
| servers talking to FTP servers, WWW clients talking to gopher |
| servers, on and on. |
| This currently works quite well for text.  But what should WWW |
| clients do as Gopher and WAIS servers begin to serve up pictures, |
| sounds, movies, spreadsheet templates, postscript files, etc.? |
| It would be a shame for each to adopt its own multimedia typing |
| system. |
| If they all adopt the MIME typing system (and as many other |
| features from MIME as are appropriate), we can step from global |
| hypertext to global hypermedia that much easier. |
+-----+
```

The fact that HTTP could reliably transport images of different formats, for example, allowed NCSA to add to HTML. MIME allowed other document formats (Word, PDF, Postscript) and other kinds of hypermedia, as well as other applications, to be part of the Web. MIME was arguably the most important extensibility mechanism in the Web.

3. Problems with application to the Web

Unfortunately, while the use of Internet Media Types for the Web added incredible power, a number of problems have arisen.

3.1. Lack of clarity

Many people are confused about the purpose of MIME in the Web, its uses, the meaning of Internet Media Types. Many W3C specifications TAG findings and Internet Media Type registrations make what are incorrect assumptions about the meaning and purposes of a Internet Media Type registration.

3.2. Differences between email and Web delivery

Some of the differences between the application contexts of email and Web delivery determine different requirements:

- o In the Web, the transfer of data is initiated differently than in email: the "messages" with labeled content are usually HTTP responses to a specific (GET) request (although the request is itself a message, GET has no content). In the most common case, then, the receiver knows more about the data before it has been sent.
- o Clients would like to know more about the content before they retrieve it. The "tagging" is often not sufficient to know, for example, "can I interpret this if I retrieve it", because of versioning, capabilities, or dependencies on things like screen size or interaction capabilities of the recipient.
- o Some content isn't delivered over the HTTP (files on local file system), or there is no opportunity for tagging (data delivered over FTP) and in those cases, some other ways are needed for determining file type.

Operating systems use (and continued to evolve) different systems to determine the 'type' of something, different from the MIME tagging and bagging:

- o 'magic numbers': in many contexts, file types could be guessed pretty reliably by looking for headers.
- o Originally MAC OS had a 4 character 'file type' and another 4 character 'creator code' for file types.
- o Windows evolved to use the "file extension" -- 3 letters (and then more) at the end of the file name -- as the initial determination of the overall type of a file. This practice has now extended to other systems.

Information about these other ways of determining type (rather than by the content-type label) were gathered for the Internet Media Type registry; those registering types are encouraged to also describe 'magic numbers', Mac file type, common file extensions. However, since there was no formal use of that information, the quality of that information in the registry is haphazard.

Finally, there was the fact that tagging and bagging might be OK for unilaterally initiated (one-way) messaging, you might want to know whether you could handle the data before reading it in and

interpreting it, but the Internet Media Types weren't enough to tell.

3.3. The Rules Weren't Quite Followed

The behavior of the community when the Internet Media Type registry was designed hasn't matched expectations:

- o Lots of file types aren't registered (no entry in IANA for file types).
- o Those that are, the registration is incomplete or incorrect (people doing registration didn't understand 'magic number' or other fields).
- o The actual content deployed or created by deployed software doesn't match the registration.

In particular, Web implementations of Internet Media Types diverged from expected behavior:

- o Browser implementors would be liberal in what they accepted, and use what looked like a file extension in the URL and/or magic number or other 'sniffing' techniques to decide file type, without assuming content-label was authoritative. This was necessary anyway for files that weren't delivered by HTTP.
- o HTTP server implementors and administrators didn't supply ways of easily associating the 'intended' file type label with the file, resulting in files frequently being delivered with a label other than the one they would have chosen if they'd thought about it, and if browsers *had* assumed content-type was authoritative. Some popular servers had default configuration files that treated any unknown type as "text/plain" (plain ext in ASCII). Since it didn't matter (the browsers worked anyway), it was hard to get this fixed.

Incorrect senders coupled with liberal readers wind up feeding a negative feedback loop based on the robustness principle ([WikiRobust], [RFC3117]).

3.4. Consequences

The result, alas, is that increased unreliability, in that

- o servers sending responses to browsers don't have a good guarantee that the browser won't "sniff" the content and decide to do something other than treat it as it is labeled

- o browsers receiving content don't have a good guarantee that the content isn't mis-labeled
- o intermediaries (gateways, proxies, caches, and other pieces of the Web infrastructure) don't have a good way of telling what the conversation means.

This ambiguity and 'sniffing' also applies to packaged content in webapps ('bagging' but using ZIP rather than MIME multipart). (NOTE: NEEDS EXPANSION, REFERENCE TO WEBAPPS)

3.5. The Down Side of Extensibility

Extensibility adds great power, and allows the Web to evolve without committee approval of every extension. For some (those who want to extend and their clients who want those extensions), this is power! For others (those who are building Web components or infrastructure), extensibility is a drawback -- it adds to the unreliability and difference of the Web experience. When senders use extensions recipients aren't aware of, implement incorrectly or incompletely, then communication often fails. With messaging, this is a serious problem, although most 'rich text' documents are still delivered in multiple forms (using multipart/alternative).

If your job is to support users of a popular browser, however, where each user has installed a different configuration of file handlers and extensibility mechanisms, MIME may appear to add unnecessary complexity and variable experience for users of all but the most popular types.

4. Additional considerations

This section notes some additional considerations.

4.1. There are related problems with charsets

MIME includes provisions not only for file 'types', but also, importantly the "character encoding" used by text types: for example, simple US ASCII, Western European ISO-8859-1, Unicode UTF8. A similar vicious cycle also happened with character set labels: mislabeled content happily processed correctly by liberal browsers encouraged more and more sites to proliferate text with mis-labeled character sets, to the point where browsers feel they *have* to guess the wrong label. (NEEDS EXPANSION)

There are sites that intentionally label content as iso-2022-jp or euc-jp when it is in fact one of the Microsoft extension charsets

(e.g., for access to circled digits. This is an intentional misuse of the definitions of the charsets themselves -- definitions which originated at the national standards body level.

4.2. Embedded, downloaded, launch independent application

The type of a document might be determined not only for entire documents "HTML" vs "Word" vs "PDF", but also to embedded components of documents, "JPEG image" vs. "PNG image". However, the use cases, requirements and likely operational impact of MIME handling is likely different for those use cases.

4.3. Additional Use Cases: Polyglot and Multiview

There are some interesting additional use cases which add to the design requirements:

- o "Polyglot" documents: A 'polyglot' document is one which is some data which can be treated as two different Internet Media Types, in the case where the meaning of the data is the same. This is part of a transition strategy to allow content providers (senders) to manage, produce, store, deliver the same data, but with two different labels, and have it work equivalently with two different kinds of receivers (one of which knows one Internet Media Type, and another which knows a second one.) This use case was part of the transition strategy from HTML to an XML-based XHTML, and also as a way of a single service offering both HTML-based and XML-based processing (e.g., same content useful for news articles and Web pages.
- o "Multiview" documents: This use case seems similar but it's quite different. In this case, the same data has very different meaning when served as two different content-types, but that difference is intentional; for example, the same data served as text/html is a document, and served as an RDFa type is some specific data.

4.4. Evolution, Versioning, Forking

The subject of format/language/type evolution is complex; this section is a little terse.

Formats and their specifications evolve over time. There are several reasons for the evolution: innovation, compatibility with other implementations, attempts to gain control.

Some times new evolutions are "compatible", although compatibility has several variations. It is part of the responsibility of the designer of a new version of a file type to try to insure both

forward and backward compatibility: new documents work reasonably (with some fallback) with old viewers and that old documents work reasonably with new viewers. In some cases this is accomplished, others not; in some cases, "works reasonably" is softened to "either works reasonably or gives clear warning about nature of problem (version mismatch)."

In MIME, the 'tag', the Internet Media Type, corresponds to the versioned series. Internet Media Types do not identify a particular version of a file format. Rather, the general idea is that the Internet Media Type identifies the family, and also how you're supposed to otherwise find version information on a per-format basis. Many (most) file formats have an internal version indicator, with the idea that you only need a new Internet Media Type to designate a completely incompatible format. The notion of an "Internet Media Type" is very coarse-grained. The general approach to this has been that the actual Media Type includes provisions for version indicator(s) embedded in the content itself to determine more precisely the nature of how the data is to be interpreted. That is, the message itself contains further information.

Unfortunately, lots has gone wrong in this scenario as well -- processors ignoring version indicators encouraging content creators to not be careful to supply correct version indicators, leading to lots of content with wrong version indicators.

Those updating an existing Internet Media Type registration to account for new versions are admonished to not make previously conforming documents non-conforming. This is harder to enforce than would seem, because the previous specifications are not always accurate to what the Internet Media Type was used for in practice.

(NOTE: MULTIPLE INCOMPATIBLE AUTHORITATIVE SPECS)

4.5. Content Negotiation

The general idea of content negotiation is when party A communicates to party B, and the message can be delivered in more than one format (or version, or configuration), there can be some way of allowing some negotiation, some way for A to communication to B the available options, and for B to be able to accept or indicate preferences.

Content negotiation happens all over. When one fax machine twirps to another when initially connecting, they are negotiating resolution, compression methods and so forth. In Internet mail, which is a one-way communication, the "negotiation" consists of the sender preparing and sending multiple versions of the message, one in text/html, one in text/plain, for example, in sender-preference order. The

recipient then chooses the first version it can understand.

HTTP added "Accept" and "Accept-language" to allow content negotiation in HTTP GET, based on Internet Media Types, and there are other methods explained in the HTTP spec.

4.6. Fragment identifiers

The Web added the notion of being able to address part of a content and not the whole content by adding a 'fragment identifier' to the URL that addressed the data. Of course, this originally made sense for the original Web with just HTML, but how would it apply to other content. The URL spec glibly noted that "the definition of the fragment identifier meaning depends on the Internet Media Type", but unfortunately, few of the Internet Media Type definitions included this information, and practices diverged greatly.

If the interpretation of fragment identifiers depends on the MIME type, though, this really crimps the style of using fragment identifiers differently if content negotiation is wanted.

5. Recommendations

This section outlines the kinds of changes needed to bring the MIME system in line with current practice and to address the problems outlined above. The purpose of this text is not to specify the exact details of how changes can be accomplished, but rather to find broad agreement.

We need a clear direction on how to make the Web more reliable, not less. We need a realistic transition plan from the unreliable Web to the more reliable one. Part of this is to encourage senders (Web servers) to mean what they say, and encourage recipients (browsers) to give preference to what the senders are sending.

We should try to create specifications for protocols and best practices that will lead the Web to more reliable and secure communication. To this end, we give an overall architectural approach to use of MIME, and then specific specifications, for HTTP clients and servers, Web Browsers in general, proxies and intermediaries, which encourage behavior which, on the one hand, continues to work with the already deployed infrastructure (of servers, browsers, and intermediaries), but which advice, if followed, also improves the operability, reliability and security of the Web.

This section outlines requirements for standards and practices

intended to address some of the difficulties. This is an early version, which mainly contains "strawman" proposals for changes. It is intended to stimulate discussion -- however, the hope is that we can get agreement about the nature of the problems and current situation before focusing in detail about possible solutions. However, having at least strawman proposals seems to be helpful. For some problems, additional changes to IETF and W3C specifications are also be advisable; the expectations are briefly outlined here.

5.1. Internet Media Type registration

Update the Internet Media Type registry and registration process.

5.1.1. MIME registry magic numbers for sniffing

Be clearer about relationship of 'magic numbers' to sniffing; review Internet Media Types already registered and update.

5.1.2. Scripting and scriptable content safety

Be clearer about requiring Security Considerations to address risks of sniffing

5.1.3. Fragment identifiers

Problem: MIME type definitions don't talk about fragment identifiers.

require definition of fragment identifier applicability; add fragID semantics

5.1.4. Application info

Problem: ((hasn't been expanded))

Could the 'applications that use this type' section to be clearer about whether the file type is frequently for embedding (plug-in) or as a separate document with auto-launch (MIME handler), or should always be downloaded? Is there a separate issue for 'auto-play on download' vs. 'ask user for permission'?

5.1.5. File extensions in registry

Problem: Sniffing needs to use file extensions too; signify which file extensions are useful for sniffing.

Be clearer about file extension use and relationship of file extensions to MIME handlers

5.2. Sniffing

Various new specifications discuss, promote or mandate the use of 'sniffing' -- using the content of the data to supplement or even override the declared content-type or charset. Update these specifications.

5.2.1. Sniffing uses Media Type magic number

Update the proposed Media Type sniffing document so that sniffing uses MIME registry for 'magic numbers'.

5.2.2. Sniffing when there are multiple different definitions

Address issue of sniffing when there are multiple independent definitions of the same MIME type.

5.2.3. Sniffing charsets

Update sniffing of charsets to use charset reference info.

5.2.4. Sniffing security uses scriptability info

If the Internet Media Type registry is more explicit about which kinds of content contain what kind of scriptability access, then the specifications for sniffing can reference the Internet Media Type registry to determine what kinds of sniffing constitute a 'privilege upgrade'.

Note that all sniffing can be a privilege upgrade, if there is a buggy recipient, although bugs can be fixed, but spec violations are a problem.

5.3. Changes to IANA processes for MIME registries

Problem: Internet Media Type registries are hard to update, and there can be different definitions of the same MIME type.

STRAWMAN: Allow commenting or easier update; not all Internet Media Type owners need or have all the information the internet needs. Wiki for Internet Media Types as well as formal registry? Ability to add comments about deployed senders, deployed content, deployed receivers.

5.4. FTP specification

Do FTP clients also change rules about guessing file types based on OS of FTP server?

5.5. Update some URI definitions

ftp, file, need sniffing, http sometimes does; data defaults to text/plain rather than sniffing. Should this info be in the URI definitions.

5.6. Changes to W3C findings, processes

Update Tag finding on authoritative metadata: is it possible to remove 'authority'?

new: MIME and Internet Media Type section to WebArch, referencing this memo

New: Add a W3C Web architecture material on MIME in HTML to W3C web site, referencing this memo

Reconsider other extensibility mechanisms (namespaces, for example): should they use MIME or something like it?

6. Acknowledgements

This document is the result of discussions among many individuals in the IETF and W3C. Special thanks to Noah Mendelsohn.

7. IANA Considerations

This document includes no specific changes to IANA registries or processes. However, it outlines several considerations for future explicit recommendations to IANA, to change Internet Media Type and Charset registries and the processes around their maintenance. IANA evaluation of the feasibility of these changed processes is required.

8. Security Considerations

This document discusses some of the security issues resulting from use (and mis-use) of MIME content types in the Web.

9. Informative References

[RFC1521] Borenstein, N. and N. Freed, "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC 1521, <<http://tools.ietf.org/html/rfc1521>>.

- [RFC1522] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Two: Message Header Extensions for Non-ASCII Text", RFC 1522, September 1993, <<http://tools.ietf.org/html/rfc1522>>.
- [RFC1945] Berners-Lee, T., Fielding, R., and H. Nielsen, "Hypertext Transfer Protocol -- HTTP/1.0", RFC 1945, May 1996, <<http://tools.ietf.org/rfc/rfc1945>>.
- [RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996, <<http://tools.ietf.org/html/rfc2046>>.
- [RFC3117] Rose, M., "On the Design of Application Protocols", RFC 3117, November 2001, <<http://tools.ietf.org/html/rfc3117>>.
- [WikiRobust] "Robustness principle", 2010, <http://en.wikipedia.org/wiki/Robustness_principle>.
- [connolly92] Connolly, D., "Global Hypermedia", Oct 1992, <<http://lists.w3.org/Archives/Public/www-talk/1992SepOct/0024.html>>.
- [mime-sniff] Barth, A. and I. Hickson, "Media Type Sniffing", May 2010, <<http://tools.ietf.org/html/draft-abarth-mime-sniff>>.

Author's Address

Larry Masinter
Adobe
345 Park Ave.
San Jose, 95110
USA

Phone: +1 408 536 3024
Email: masinter@adobe.com
URI: <http://larry.masinter.net>

