Network Working Group
Request for Comments: 3650
Category: Informational

S. Sun
L. Lannom
B. Boesch
CNRI

November 2003

Handle System Overview

Status of this Memo

Copyright Notice

IESG Note

Several groups within the IETF and IRTF have discussed the Handle System and its relationship to existing systems of identifiers. The IESG wishes to point out that these discussions have not resulted in IETF consensus on the described Handle System, nor on how it might fit into the IETF architecture for identifiers. Though there has been discussion of handles as a form of URI, specifically as a URN, these documents describe an alternate view of how namespaces and identifiers might work on the Internet and include characterizations of existing systems which may not match the IETF consensus view.

Abstract

This document provides an overview of the Handle System in terms of its namespace and service architecture, as well as its relationship to other Internet services such as DNS, LDAP/X.500, and URNs. The Handle System is a general-purpose global name service that allows secured name resolution and administration over networks such as the Internet. The Handle System manages handles, which are unique names for digital objects and other Internet resources.

Table of Contents

1. Introduction

This document provides an overview of the Handle System, a distributed information system designed to provide an efficient, extensible, and secured global name service for use on networks such as the Internet. The Handle System includes an open protocol, a namespace, and a reference implementation of the protocol. The protocol enables a distributed computer system to store names, or handles, of digital resources and resolve those handles into the information necessary to locate, access, and otherwise make use of the resources. These associated values can be changed as needed to reflect the current state of the identified resource without changing the handle. This allows the name of the item to persist over changes of location and other current state information. Each handle may have its own administrator(s) and administration can be done in a distributed environment. The Handle System supports secured handle resolution. Security services such as data confidentiality, data integrity, and non-repudiation are provided upon client request.

The Handle System provides a confederated name service that allows any existing local namespace to join the global handle namespace by obtaining a unique Handle System naming authority. Local names and their value-binding(s) remains intact after joining the Handle System. Any handle request to the local namespace may be processed

by a service interface speaking the Handle System protocol. Combined with the unique naming authority, any local name is guaranteed unique under the global handle namespace.

There are several services used today to provide name service for Internet resources. Among these, the Domain Name System (DNS) [2,3] is the most widely used. DNS is designed "to provide a mechanism for naming resources in such a way that the names are mappable into IP addresses and are usable in different hosts, networks, protocol families, internets, and administrative organizations" [3]. The growth of the Internet has raised demands for various extensions to DNS. There are also attempts to use DNS as a general-purpose resource naming system. However, the importance of DNS in basic network routing has led to great caution in implementing any DNS extension or overloading the DNS for general-purpose resource naming. An additional factor which argues against using DNS as a general-purpose naming service is the DNS administrative model. DNS names are typically managed by the network administrator(s) at the DNS zone level. There is no provision for per-name administrative structure and no facilities for anyone other than the network administrator to create or manage DNS names. This is appropriate for domain name administration, but less so for general-purpose resource naming.

The Handle System has been designed from the start to serve as a general-purpose naming service. It is designed to accommodate very large numbers of entities and to allow distributed administration over the public Internet. The Handle System data model allows access control to be defined at the level of each of the data values associated with a given handle. Each handle can further define its own set of administrators that are independent from the network or host administrator.

Traditional URLs (Uniform Resource Locators) [4] allow certain Internet resources to be named as a combination of a DNS name and local name. The local name may be a local file path, or a reference to some local service (e.g., a cgi-bin script). This combination of a DNS name and a local name provides a flexible administrative model for naming and managing individual Internet resources. However, the URL practice also has some key limitations. Most URL schemes (e.g., http) are defined for resolution only. Any URL administration has to be done either at the local host, or via some other network service such as NFS. Using a URL as a name typically ties the Internet resource to its current network location. For example, a URL will be tied to its local file path when the file path is part of the URL. When the resource moves from one location to another for whatever reason, the URL breaks. It is especially difficult to work around

this problem when the reason for the location change is change in ownership of an asset, as ownership is generally reflected in the domain name.

The Handle System is designed to overcome these limitations and to add significant functionality. Specifically, the Handle System is designed with the following objectives:

- Uniqueness: Every handle is globally unique within the Handle System.

- Persistence: Handles may be used as persistent identifiers for Internet resources. A handle does not have to be derived from the entity that it names. While an existing name, or even a mnemonic, may be included in a handle for convenience, the only operational connection between a handle and the entity it names is maintained within the Handle System. This of course does not guarantee persistence, which is a function of administrative care. But it does allow the same name to persist over changes of location, ownership, and other state conditions. For example, when a named resource moves from one location to another, the handle may be kept valid by updating its value in the Handle System to reflect the new location.

- Multiple Instances: A single handle can refer to multiple instances of a resource, at different and possibly changing locations in a network. Applications can take advantage of this to increase performance and reliability. For example, a network service may define multiple entry points for its service with a single handle so as to distribute the service load.

- Multiple Attributes: A single handle can refer to multiple attributes of a resource, including associated services, available through any method at different and possibly changing network locations. Handles can thus be used as persistent entry points into an evolving world of services associated with identified resources.

- Extensible Namespace: Existing local namespaces may join the handle namespace by acquiring a unique handle naming authority. This allows local namespaces to be introduced into a global context while avoiding conflict with existing namespaces. Use of naming authorities also allows delegation of service, both resolution and administration, to a local handle service.

- International Support: The handle namespace is based on Unicode 3.0 [17], which includes most of the characters currently used around the world. This allows handles to be used in any native environment. The handle protocol mandates UTF-8 [5] as the encoding used for handles.

- Distributed Service Model: The Handle System defines a hierarchical service model such that any local handle namespace may be serviced by a corresponding local handle service, by the global service, or by both. The global service, known as the Global Handle Registry, can be used to dispatch any handle service request to the responsible local handle service. The distributed service model allows replication of any given service into multiple service sites, and each service site may further distribute its service into a cluster of individual servers. (Note that local here refers only to namespace and administrative concerns. A local handle service could in fact have many service sites distributed across the Internet.)

- Secured Name Service: The Handle System allows secured name resolution and administration over the public Internet. The Handle System protocol defines standard mechanisms for both client and server authentication, as well as service authorization. It also provides security options to assure data integrity and confidentiality.

- Distributed Administration Service: Each handle may define its own administrator(s) or administrator group(s). Ownership of each handle is defined in terms of its administrator or administrator groups. This, combined with the Handle System authentication protocol, allows any handle to be managed securely over the public network by its administrator at any network location.

- Efficient Resolution Service: The handle protocol is designed to allow highly efficient name resolution performance. To avoid resolution being affected by computationally costly administration service, separate service interfaces (i.e., server processes and their associated communication ports) for handle name resolution and administration may be defined by any handle service.

This document provides an overview of the handle namespace and service architecture. It also compares the Handle System with other existing Internet services, protocols, and specifications (e.g., DNS [2, 3], URLs [4], X.500/LDAP [6,7,8], and URN [9,10]). Details of the handle system data and service model, as well as its communication protocol, are specified in separate documents. They

can be found under the Handle System website at http://www.handle.net.

2. Motivations

Since there are a number of name related projects in the Internet community, it is worth defining exactly where we believe the Handle System fits. Unfortunately, that is particularly hard because the other primary naming schemes either take an abstract services approach (e.g., URI/URN), or an approach to name resolution absent of a self-contained framework for reliable yet distributed administration of the underlying databases (e.g., DNS). This makes categorizing the Handle System difficult.

The Handle System crosses boundaries. Looked at as a name resolution system, it might be compared to DNS. If used to implement a URI/URN namespace, it could be used with any URI/URN scheme. If used for distributed information updates and administration, it could be considered a simplified-version of a distributed database system.

It is probably best to view the Handle System as a name-attribute binding service with a specific protocol for securely creating, updating, maintaining, and accessing a distributed database. It is designed to be an enabling service for secured information and resource sharing over networks such as the public Internet. Applications of the Handle System could include meta-data services for digital publications, identity management services for virtual identities, or any other applications that require resolution and/or administration of globally unique identifiers.

In the spirit of exploration, the Handle System has been designed to have high performance for name resolution and to push the boundaries of distributed access control and administration. Unlike most conventional systems (even distributed systems) that are designed to have a relatively small number of broadly empowered administrators, the Handle System allows extremely fine granularity of administrative control. It has a unique self-contained administrative framework that de-couples the ownership of each handle from the system administrators and allows access control to be defined for each handle value.

It should be noted, that as with all real systems, the Handle System is a compromise between a number of technical and practical concerns. There are also different opinions within the IETF on where the Handle System fits in relation to other existing Internet name services. It is with the goal of exposing a broader community to the concepts, approach, specific decisions, tradeoffs and results that we are writing this RFC.

3. Handle Namespace

Every handle consists of two parts: its naming authority, otherwise known as its prefix, and a unique local name under the naming authority, otherwise known as its suffix:

<Handle> ::= <Handle Naming Authority> "/" <Handle Local Name>

The naming authority and local name are separated by the ASCII character "/". The collection of local names under a naming authority defines the local handle namespace for that naming authority. Any local name must be unique under its local namespace. The uniqueness of a naming authority and a local name under that authority ensures that any handle is globally unique within the context of the Handle System.

For example, "10.1045/january99-bearman" is a handle for an article published in D-Lib magazine [12]. Its naming authority is "10.1045" and its local name is "january99-bearman". The handle namespace can be considered a superset of many local namespaces, with each local namespace having a unique naming authority under the Handle System. The naming authority identifies the administrative unit of creation, although not necessarily continuing administration, of the associated handles. Each naming authority is guaranteed to be globally unique within the Handle System. Any existing local namespace can join the global handle namespace by obtaining a unique naming authority so that any local name under the namespace can be globally referenced as a combination of the naming authority and the local name as shown above.

Naming authorities under the Handle System are defined in a hierarchical fashion resembling a tree structure. Each node and leaf of the tree is given a label that corresponds to a naming authority segment. The parent node notifies the parent naming authority of its child nodes. Unlike DNS, handle naming authorities are constructed left to right, concatenating the labels from the root of the tree to the node that represents the naming authority. Each label is separated by the octet used for ASCII character "." (0x2E). For example, a naming authority for the National Digital Library Program ("ndlp") at the Library of Congress ("loc") is defined as "loc.ndlp".

Each naming authority may have many child naming authorities registered underneath. Any child naming authority can only be registered by its parent after its parent naming authority has been registered. However, there is no intrinsic administrative relationship between the namespaces represented by the parent and child naming authorities. The parent namespace and its child

namespaces may be served by different handle services, and they may or may not share any administration privileges.

Handles may consist of any printable characters from the Universal Character Set (UCS-2) of ISO/IEC 10646, which is the exact character set defined by Unicode v3.0 [17]. The UCS-2 character set encompasses most characters used in every major language written today. To allow compatibility with most of the existing systems and to prevent ambiguity among different encodings, the Handle System protocol mandates UTF-8 to be the only encoding used for handles. The UTF-8 encoding preserves any ASCII encoded names so as to allow maximum compatibility with existing systems without causing naming conflict. Some encoding issues over the global namespace and the choice of UTF-8 encoding are discussed in [13].

By default, handles are case sensitive. However, any individual handle service may define its namespace such that ASCII characters within any handle under that namespace are case insensitive.

4. Handle System Architecture

The Handle System defines a hierarchical service model. The top level consists of a single handle service, known as the Global Handle Registry (GHR). The lower level consists of all other handle services, generically known as Local Handle Services (LHS).

The Global Handle Registry can be used to manage any handle namespace. It is unique among handle services only in that it provides the service used to manage naming authorities, all of which are managed as handles. The naming authority handle provides information that clients can use to access and utilize the local handle service for handles under the naming authority.

Local Handle Services are intended to be hosted by organizations with administrative responsibility for handles under certain naming authorities. A Local Handle Service may be responsible for any number of local handle namespaces, each identified by a unique naming authority. The Local Handle Service and its responsible set of local handle namespaces must be registered with the Global Handle Registry.

One important aspect of the Handle System is its distributed architecture. The Handle System as a whole consists of a number of individual handle services. Each of these services may consist of one or more service sites. Each service site is a complete replication of every other site in the service in terms of handle resolution. Each service site may consist of one or more handle servers. All handles, and hence all handle requests, directed at a given service site will be evenly distributed across these handle

servers. The Handle System as a whole may consist of any number of handle services. There are no design limits on the number of handle services or on the number of sites which make up each service, nor are there any limits on the number of servers that make up each site. Replication among any service site does not require that each site contain the same number of servers. In other words, while each site will have the same replicated set of handles, each site may allocate that set of handles across a different number of servers. This distributed approach is intended to aid scalability, accommodate any large-scale of operation, and mitigate problems of single point failure.

Figure 3.1 illustrates a potential handle service that consists of two service sites: one located on the U.S. east coast and the other on the U.S. west coast. The east coast service site consists of four server computers. The west coast service site, with more powerful computers deployed, decides two servers will suffice. The number of service sites for any handle service, as well as the number of servers that are used by any service site, may be added or removed dynamically depending on the service requirement.

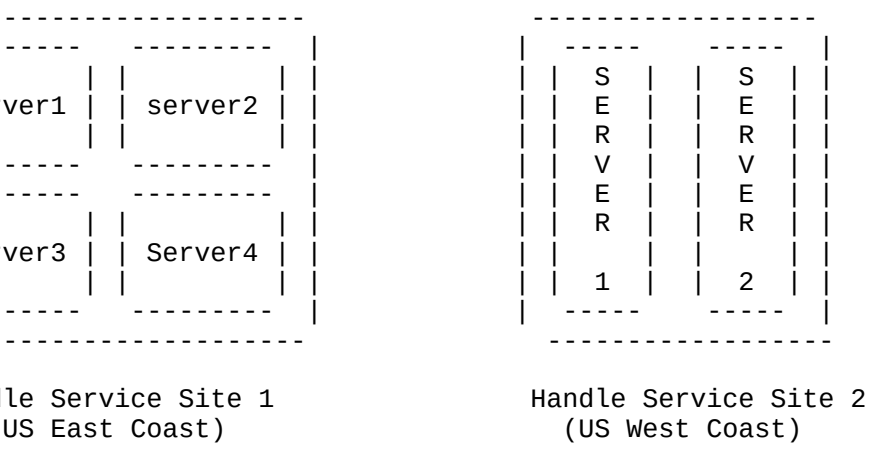

Figure 3.1: Handle service configured with two service sites

Each handle service manages a distinct sub-namespace under the Handle System. Namespaces under different handle services may not overlap. The sub-namespace typically consists of handles under a number of naming authorities. The handle service is called the "home" service of these naming authorities and is the only one that provides resolution and administration service for handles under these naming authorities. Before resolving a handle, a client has to determine the "home" service of the handle in question. The "home" service of each handle is the "home" service of its naming authority and is

registered at the Global Handle Registry. Clients can find the "home" service for each handle by querying the naming authority handle at the Global Handle Registry.

The Global Handle Registry maintains naming authority handles. Each naming authority handle maintains the service information that describes the "home" service of the naming authority. The service information lists the service sites of the given handle service, as well as the interface to each handle server within each site. To find the "home" service for any handle, a client can query the Global Handle Registry for the service information associated with the corresponding naming authority handle. The service information provides the necessary information for clients to communicate with the "home" service.

Figure 3.2 shows an example of a typical handle resolution process. In this case, the "home" service is a Local Handle Service. The client is trying to resolve the handle "10.1045/july95-arms" and has to find its "home" service from the Global Handle Registry. The "home" service can be found by sending a query to the Global Handle Registry for the naming authority handle for "10.1045". The Global Handle Registry returns the service information of the Local Handle Service that is responsible for handles under the naming authority "10.1045". The service information allows the client to communicate with the Local Handle Service to resolve the handle "10.1045/july95-arms".

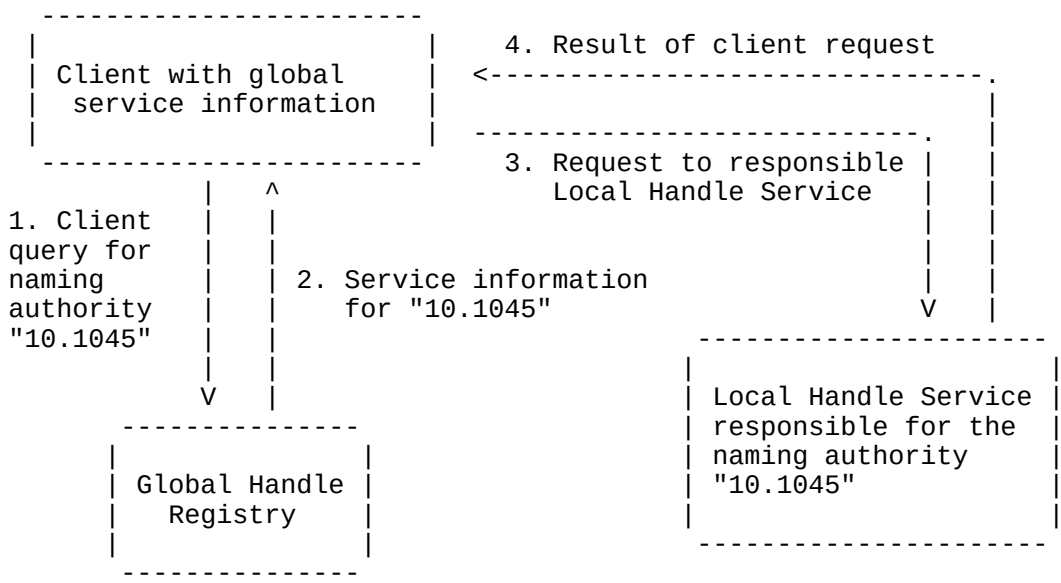

Figure 3.2: Handle resolution starting with global

To improve resolution performance, any client may choose to cache the service information returned from the Global Handle Registry and use it for subsequent queries. A separate handle caching server, either stand-alone or as a piece of a general caching mechanism, may also be used to provide shared caching within a local community. Given a cached resolution result, subsequent queries of the same handle may be answered locally without contacting any handle service. Given cached service information, clients can send their requests directly to the correct Local Handle Service without contacting the Global Handle Registry.

5. Handle System Security

The Handle System provides handle resolution and administration service over networks such as the public Internet. Each handle can be assigned a set of values. Clients use the handle resolution service to resolve any handle into its set of values. Each value has a data type and a unique value index. Clients can query for specific handle values based on data type or value index.

The handle administration service answers requests from clients to manage handles. These include adding handles, deleting handles or updating their values. It also manages naming authorities via naming authority handles. Each handle can have its own administrator(s), and each administrator can be granted a certain set of permissions.

The handle system authentication protocol authenticates the handle administrator before fulfilling any administrative request.

The Handle System provides security services such as client and server authentication, data confidentiality and integrity, and non-repudiation. By default, handle resolution does not require any client authentication. However, resolution requests for confidential data assigned to any handle (by its administrator), as well as any administration requests (e.g., adding or deleting handle values) require authentication of the client for proper authorization. The server will decide, during the authorization process, whether or not the client has permission to access those confidential handle values, or has permission to add or update handles and handle values. When authentication is required, the handle server will issue a challenge to the requesting client before carrying out the client's request. To satisfy the authentication requirement, the client must send back the correct response identifying itself as a qualified administrator. The handle server will respond to the initial request only after successful authentication of the client. Handle clients may choose to use either secret key or public key cryptography for authentication. Handle System authentication can also be carried out via third party authentication services. To ensure data integrity, clients may request digitally signed responses from any handle server. They may also set up secured communication sessions with handle servers so that any exchanged information can be encrypted (for data confidentiality) using a session key. Handle servers can also provide confidentiality by encrypting the handle data before sending it to the client.

The Handle System provides service options for secured information exchange between the client and server. This does not, of course, guarantee the truthfulness of handle values. Incorrect values assigned to any handle by its administrator may very well mislead clients. On the other hand, a handle value may contain references to other handle values to provide additional credentials. For example, a handle value R (e.g., a claim) may contain a reference to some other handle value that contains the digital signature (from a creditable source) upon the value R. Clients who trust the signature could then trust the handle value R.

6. The Handle System and other Internet Services

There are a number of existing and proposed Internet identifier services or specifications that, by design or intent, cover some of the functionalities proposed for the Handle System. This section briefly reviews them in relationship to the Handle System.

6.1. Domain Name Service (DNS)

The Domain Name Service, or DNS, was originally designed and is heavily used for mapping domain names into IP Addresses for network routing purposes. RFC 1034 [2] and RFC 1035 [3] provide detailed descriptions of its design and implementation. The growth of the Internet has increased demands for various extensions to DNS, even its possible use as a general purpose resource naming system. However, any such use has the potential to slow down the network address translation and/or affect its effectiveness in network routing. DNS implementations typically do not scale well when a large amount of data is associated with any particular DNS name. It is therefore generally considered inappropriate to use DNS as a general-purpose naming service.

An additional factor that argues against using DNS as a general-purpose naming service is the DNS administrative model. DNS names are typically managed by the network administrator(s) at the DNS zone level. There is no provision for a per-name administrative structure. No facilities are provided for anyone other than network administrators to create or manage DNS names. This is appropriate for domain name administration but less so for general-purpose name administration.

The Handle System differs from DNS in its distributed administration and service model, as well as its security features. The handle system protocol includes security options to assure confidentiality and integrity during data transmission. Each handle can have its own administrator, independent from the server administrator. The handle system protocol allows any handle administrator to manage his or her handles securely over the public network. Additionally, the Handle System service model allows any of its service sites to dynamically configure its service distribution among a cluster of servers to accommodate increased service requests. This also allows less powerful computers to be used together to support any arbitrarily large number of handles.

6.2. Directory Services (X.500/LDAP)

X.500 [6] is the OSI Directory Standard defined by the ISO and the ITU. It is designed "to provide a white pages service that would return either the telephone numbers or X.400 O/R addresses of people", and is "concerned mainly with providing the name server service for Open Systems Interconnection (OSI) applications" [7]. X.500 defines a hierarchical data and information model with a set of protocols to allow global name lookup and search. The protocol, however, has proved difficult to implement and there has been difficulty in getting "client access integrated into existing

products" [14]. LDAP (Lightweight Directory Access Protocol) [8] has overcome many of these difficulties by making the protocol simpler and easier to implement. Some concern remains, however, that as LDAP is emerging from a local directory access protocol (LDAP v2) into a distributed service protocol (LDAP v3), it faces many issues not addressed in its original design, resulting in new complications.

The fundamental difference between a name resolution service such as the Handle System, and a directory service such as LDAP, is search capability. The added functionality of being able to search a directory service necessarily carries with it added complexity, thus affects its efficiency. A pure name service, such as the Handle System, can be designed solely around efficient resolution of known items without addressing functions and data structures required for discovery of unknown items based on incomplete criteria.

Directory services, such as LDAP or WHOIS++ [15,16], may be used in tandem with the Handle System to provide reverse lookup service. Existing corporate directory services, for example, could provide interfaces to both services. The Handle System interface would provide a highly efficient name resolution service. The directory service interface would provide extended search capability. Handles could also be used in LDAP service referral. For example, an LDAP service may be referenced as a handle. Doing so will make the reference persistent overtime, independent of location change.

6.3. Uniform Resource Identifier (URI)/Uniform Resource Name(URN)

Uniform Resource Identifier (URI) [23] defines a uniform, yet extensible naming mechanism for identifying Internet resources in web applications. Uniform Resource Name (URN) [11], a subset of URI, defines a namespace registration mechanism for persistent namespaces under URI. URI/URN represents most of the Internet name services used in web applications. This section discusses the relationship of the Handle System to URI/URN and how applications may utilize the Handle System within the URI/URN context.

The Handle System provides a general-purpose name service for the Internet. Like DNS or X.500 directory service, the Handle System defines its namespace outside of any URI/URN namespace. Handles can be transcribed and resolved directly, without any URI/URN scheme as a prefix. For example, a library application may resolve the handle "10.1045/july95-arms" directly into its set of handle values. No URI/URN scheme will be needed in this case.

The Handle System may be used for applications that require a persistent name service. The Handle System provides the necessary mechanisms to allow persistent names to be registered as handles.

Specific naming authorities may be defined to host those handles designed to be persistent. However, the persistence of handles depends more on administrative policies than the technology itself. Such policies are beyond the Handle System service, as described in this set of documents.

On the other hand, the Handle System can also be used for applications where persistent names are not required. Such handles may have a short life-time and they may also be used to identify different objects at different times.

Different web applications may be developed using the Handle System as the underlying name service. Each of these applications may define its own URI/URN namespace for its application needs. For example, application FOO may have a URI namespace "foo:" registered to identify any FOO services on the web. In the mean time, application BAR may have a URN namespace "URN:BAR" registered to identify any BAR object that needs a persistent name. Both FOO and BAR applications may use handles (under their respective naming authority) in naming and resolving to services and/or objects. This is similar in DNS, where there are different URI schemes (e.g., "telnet", "ftp", "mailto", etc.) defined for different applications, all using the DNS service.

The IETF and IRTF have discussed the Handle System in the realm of URI-related work. There are different opinions on whether the Handle System will fit into a specific URI or URN namespace. There are also concerns on where the Handle System fits in relation to other existing name services on the Internet. Such discussions are out of the scope of this document.

7. Security Considerations

This section is meant to inform people of security limitations of the Handle System, as well as precautions that should be taken by application developers, service providers, and Handle System clients. Specific security considerations regarding the Handle System protocol [21], as well as its data and service model [22], are addressed in separate documents.

7.1. General Security Practice

The security of the Handle System depends on both client and server host security at every step in the transaction. It assumes the client host has not been tampered with and that client software will reliably convey the received data to the client. The client of any handle service must also assume that any handle servers involved have not been compromised. To trust the Global Handle Registry is to

believe that the Global Handle Registry will correctly direct the client request to the responsible Local Handle Service. To trust a Local Handle Service is to believe that the Local Handle Service will correctly return the data that was assigned to the handle by its administrator. A Local Handle Service typically supports a set of naming authorities. Thus, trusting a Local Handle Service would imply trusting those naming authorities.

The integrity of the Handle System depends heavily on the integrity of the global service information. Invalid global service information may mislead clients into inappropriate Local Handle Services. It may also allow attackers to forge server signatures. The Global Handle Registry must take extreme caution in protecting the global service information and the public key pair used to sign the global service information. Client applications should only accept the global service information from the Global Handle Registry. They should check its integrity upon each update.

For efficiency reasons, handle servers will not generate or return a digital signature for every service response, unless specifically requested by clients. To assure data integrity, clients must explicitly ask the server to return the digital signature. To protect sensitive data from exposure, clients may establish a communication session with the server and ask the server to encrypt any data using the session key.

7.2. Privacy Protection

By default, most handle data stored in the Handle System is publicly accessible, unless otherwise specified by the handle administrator. Handle administrators must pay attention when adding handle values that contain private information. They may choose to mark these handle values readable only by the handle administrator(s), or to store these as encrypted handle values, so that these values can only be read within a controlled audience.

Log files generated by the handle server are another vulnerable point where client privacy may be under attack. Operators of handle servers must protect such information carefully.

7.3. Caching and Proxy Servers

Besides performance gains and other value-added services, both proxy and caching servers present themselves as men-in-the-middle, and as such are vulnerable to man-in-the-middle attacks. It is important to know that proxy and caching servers are not part of any handle service. They are clients of the Handle System. Service responses from proxy and caching servers cannot be authenticated via the Handle

System protocol. The trust between the client and its immediate proxy/caching server has to be setup independently, regardless of the number of proxy/caching servers that are in the middle of the communication path.

By using proxy and caching servers, clients assume that the servers will submit their requests and relay any responses from the Handle System without mishandling any of the contents. They also assume that the servers will protect any sensitive information on their behalf.

Proxy and caching server operators should protect the systems on which such servers are running as they would protect any system that contains or transports sensitive information. In particular, log information gathered at proxies often contain highly sensitive personal information, and/or information about organizations. Such information should be carefully guarded, and appropriate guidelines for their use developed and followed.

Caching servers provide additional potential vulnerabilities because the contents of the cache represent an attractive target for malicious exploitation. Potential attacks on the cache can reveal private data for a handle user, or information still kept after a user believes that they have been removed from the network. Therefore, cache contents should be protected as sensitive information.

7.4. Mirroring

Handle System clients should be aware of possible delays in content replication among mirroring sites. They should consider sending their request to the primary service site for any time-sensitive data. Selection of mirroring sites by service administrators must be done carefully. Each mirroring site must follow the same security procedures in order to ensure data integrity. Software tools may be applied to ensure data consistency among mirroring sites.

7.5. Denial of Service (DoS)

As with any public service, the Handle System is subject to denial of service attacks. No general solutions are available to protect against such attacks in today’s technology. Server implementations may be developed to be aware of such attacks and notify administrators when they happen. Stateless cookies [19, 20] are one means of mitigating some of the effects of DoS attacks on hosts that perform authentication, integrity, and encryption services. Server

implementations, moreover, need to be upgradeable to take advantage of new security technologies, including anti-DoS technologies as these become available.

8. History of the Handle System

The Handle System was originally conceived and developed at CNRI as part of an overall digital object architecture. The first public implementation was created at CNRI in the fall of 1994 in an effort led by David Ely. The overall digital object architecture, including the Handle System, was later described in a paper by Robert Kahn and Robert Wilensky [1] in 1995. Development continued at CNRI as part of the Computer Science Technical Reports (CSTR) project, funded by the Defense Advanced Projects Agency (DARPA) under Grant Number MDA-972-92-J-1029 and MDA-972-99-1-0018. One aspect of this early digital library project, which was also a major factor in the evolution of the Networked Computer Science Technical Reference Library (NCSTRL) [18] and related activities, was to develop a framework for the underlying infrastructure of digital libraries.

Early adopters of the Handle System included the Library of Congress, the Defense Technical Information Center (DTIC), and the International DOI Foundation (IDF). Feedback from these organizations as well as NCSTRL, other digital library projects, and related IETF efforts as mentioned above, have all contributed to the evolution of the Handle System. The current status and available software, for both client and server, can be found at http://www.handle.net.

9. Acknowledgements

This work is derived from the earlier versions of the Handle System implementation. Design ideas are based on those discussed within the Handle System development team, including David Ely, Charles Orth, Allison Yu, Sean Reilly, Jane Euler, Catherine Rey, Stephanie Nguyen, Jason Petrone, and Helen She. Their contributions to this work are gratefully acknowledged.

The authors also thank Russ Housley (housley@vigilsec.com), Ted Hardie (hardie@qualcomm.com), and Mark Baugher (mbaugher@cisco.com) for their extensive review and comments, as well as recommendations received from other members of the IETF/IRTF community.

10. References and Bibliography

[1] Kahn, R. and R. Wilensky, "A Framework for Distributed Digital Object Services", D-Lib Magazine, 1995.

[2] Mockapetris, P., "Domain Names - Concepts and Facilities", STD 13, RFC 1034, November 1987.

[3] Mockapetris, P., "Domain Names - Implementation and Specification", STD 13, RFC 1035, November 1987.

[4] Berners-Lee, T., Masinter, L. and M. McCahill, "Uniform Resource Locators (URL)", RFC 1738, December 1994.

[5] Yergeau, F., "UTF-8, a transformation format of Unicode and ISO 10646", RFC 2044, October 1996.

[6] ITU-T Rec. X.500, "The Directory: Overview of Concepts, Models, and Services", 1993.

[7] D. W. Chadwick, "Understanding X.500 - The Directory", Chapman & Hall ISBN: 0-412-43020-7.

[8] Wahl, M., Howes, T. and S. Kille, "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997.

[9] Sollins, K. and L. Masinter, "Functional Requirements for Uniform Resource Names", RFC 1737, December 1994.

[10] Sollins, K. "Architectural Principles of Uniform Resource Name Resolution", RFC 2276, January 1998.

[11] IETF Uniform Resource Names (URN) Working Group, April 1998.

[12] D-Lib Magazine, http://www.dlib.org

[13] Sam X. Sun, "Internationalization of the Handle System - A Persistent Global Name Service", Proceeding of 12th International Unicode Conference, April 1998.

[14] D. Goodman, C. Robbins, "Understanding LDAP & X.500", August 1997.

[15] Deutsch P., Schoultz R., Faltstrom P. and C. Weider, "Architecture of the WHOIS++ service", RFC 1835, August 1995.

[16] Weider, C., Fullton, J. and S. Spero, "Architecture of the Whois++ Index Service", RFC 1913, February 1996.

[17] The Unicode Consortium, "The Unicode Standard, Version v3.0", Addison-Wesley Pub Co; ISBN: 0201616335.

[18] The Networked Computer Science Technical Reports Library (NCSTRL), http://www.ncstrl.org/

[19] Karn, P. and W. Simpson, "Photuris: Session-Key Management Protocol", RFC 2522, March 1999.

[20] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[21] Sun, S., Reilly, S. and L. Lannom, "Handle System Namespace and Service Definition", RFC 3651, November 2003.

[22] Sun, S., Reilly, S., Lannom, L. and J. Petrone, "Handle System Protocol (ver 2.1) Specification", RFC 3652, November 2003.

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

11. Authors' Addresses

Sam X. Sun
Corporation for National Research Initiatives (CNRI)
1895 Preston White Dr., Suite 100
Reston, VA 20191

Phone: 703-262-5316
EMail: ssun@cnri.reston.va.us

Larry Lannom
Corporation for National Research Initiatives (CNRI)
1895 Preston White Dr., Suite 100
Reston, VA 20191

Phone: 703-620-8990
EMail: llannom@cnri.reston.va.us

Brian Boesch
Corporation for National Research Initiatives (CNRI)
1895 Preston White Dr., Suite 100
Reston, VA 20191

Phone: 703-262-5316
EMail: bboesch@cnri.reston.va.us

12. Full Copyright Statement

Acknowledgement

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