Domain Name System Wildcards in Top-Level Domain Zones

Scott Hollenbeck <shollenbeck@verisign.com>
Matt Larson <mlarson@verisign.com>
VeriSign Naming and Directory Services
VeriSign, Inc.
9 September 2003

TABLE OF CONTENTS

1 Introduction

This white paper describes suggested guidelines for deploying a standard "wildcard" in top-level domain (TLD) zones. Consistent with RFC 1034 [1], Domain Name System (DNS) wildcards allow a zone administrator to synthesize resource records in response to queries that do not match an otherwise existing domain name. Consider the domain name "bookstoore.com": a DNS query for the address of a non-existent domain name like "bookstoore.com" produces a Name Error response instead of an authoritative address. A wildcard response for a query for "bookstoore.com" results in a response containing the address of a "response" server that responds to various applications, such as web service. The "response" server sends a dynamically generated web page to the browser that provides the user with multiple search options for similar domain names (such as "bookstore.com"). The user, therefore, is given information that helps them reach their intended destination instead of being frustrated by a cryptic application error message.

Several TLD administrators[1] already support wildcard functionality in their zones, demonstrating that the concept works in practice. The applications provided by these administrators to support wildcard functionality vary, but in all cases the administrators provide a web page to inform human web users that they have reached a destination as a result of attempting to resolve a non-existent domain name. In most cases, the web page informs the user that the domain is available for registration. In one case the web page helps the user find web sites associated with delegated subdomains.

This paper will first describe wildcards as they are specified by the DNS protocol to provide a common understanding of the mechanics for their use. Second, it provides considerations for zone administrators who wish to deploy a DNS wildcard in a top-level domain zone. Specific guidelines for wildcard implementation are noted throughout this document as numbered paragraphs that start with the letter "G", for "Guideline", followed by a revision number and the guideline number. For example, the first guideline is numbered "G1.1", the second is numbered "G1.2", and so on.

While the guidelines described in this paper are presented in the context of top-level domain zones, they can be applied to other zones as well. Any zone administrator who wishes to associate an information server with a zone may find the guidelines useful.

2 Wildcards in the DNS

The standard DNS protocol supports wildcards, which allow a uniform response to any queries for non-existent domain names. A zone's authoritative servers synthesize resource records in response to queries matching wildcard entries in the zone: when an

[1] The zones for .cc, .cx, .io, .mp, .museum, .nu, .ph, .td, .tk, .tv, and .ws support wildcard functionality.

authoritative server receives a query matched by a wildcard entry, the server creates the appropriate resource records for the response in real time.

2.1 Wildcard Resource Record Format

DNS wildcard resource record syntax is defined in section 4.3.3 of RFC 1034 [1]. A wildcard entry, like any other resource record, has a specific type, such as A (address), MX (mail exchange), etc. When a zone's authoritative server receives a query for a domain name that doesn't exist in that zone, and the type of the query matches the type of a wildcard entry present in the zone, the server synthesizes a response from the matching wildcard entry.

A wildcard is represented with an asterisk in the owner name field[2] of a resource record. For example, a wildcard address record in the *.example*[3] zone might look like this:

```
*.example. 300 IN A 192.0.2.16
```

Any queries of type A received by an authoritative server for the *.example* zone for domain names not present in the zone would result in a response containing the record shown above. In the response, the owner name of the record would be changed to match the queried domain name from the original query. The owner name of records in the Answer section of a DNS message containing a response always matches the owner name of the original query, which is contained in the Question section of the message. DNS queries and responses use the same packet format, which is called a "DNS message" and has five sections. For details, see section 3.7 of RFC 1034 [1].

This paper uses the term "response server" to describe the server (or servers) associated with the address returned in response to a query matching a wildcard address record (or records) in a TLD zone. Note that a response server is distinct from a "name server" authoritative for a TLD zone.

2.2 Wildcard Responses vs. Non-wildcard Responses

When an authoritative server receives a query that is matched by the wildcard, it synthesizes records for the response as described above. It is important to note that this response, though generated as a result of a wildcard, does not differ from a non-wildcard-related response. The recipient cannot determine the presence of a wildcard entry in a zone from only a single response generated as a result of that wildcard. No special wildcard processing, nor even knowledge of the existence of wildcards in the DNS protocol, is required on stub or recursive resolvers. Resolvers are DNS clients and the

[2] In a DNS context, "owner" refers simply to the domain name that a particular resource record pertains to.
[3] RFC 2606 calls for the reservation of several top-level domains for example purposes, including the appropriately named *.example*.

roles of different kinds of resolvers are described in RFC 1034.

The only way for a querier to determine the presence of a wildcard in a zone is to query specifically for the wildcard by domain name. For example, a query for domain name "*.museum", type A (address), class IN (Internet) sent to an authoritative server for the *.museum* zone produces a response confirming the presence of a wildcard in that zone:

```
*.museum. 86400 IN A 195.7.77.20
```

2.3 Compatibility with DNSSEC

RFC 2535 [2] describes extensions to the DNS protocol to authenticate DNS data using cryptography. These extensions are often referred to as "DNSSEC". Wildcards have always been and continue to be compatible with DNSSEC. The SIG record includes a field listing the number of labels in a domain name specifically to accommodate wildcards; see RFC 2535 section 4.1.3. Additional information about wildcards in DNSSEC can be found in RFC 2535 section 5.3.

2.4 Time to Live (TTL) Considerations

Zone administrators should consider choosing an appropriate time to live (TTL) value for wildcard entries. The TTL of wildcard entries should not be too long. Responses synthesized as a result of a wildcard are cached just like other responses because these synthesized responses are indistinguishable from other responses. Name server operators should expect an increase in the number of cached positive responses and a corresponding increase in cache size and may need to address cache size issues. For example, some older recursive name server software does not gracefully handle the situation when the cache size exceeds available memory. A shorter TTL means individual entries are retained for a shorter period, which reduces overall resource requirements. In addition, a shorter TTL gives the zone administrator more flexibility to change the wildcard entry; after the entry is changed, its previous value disappears faster from caches throughout the Internet. The TTL of the wildcard thus becomes the time it takes for new names to be seen after they have been added to the zone.

G1.1. Zone administrators should consider choosing an appropriate time to live (TTL) value for wildcard entries. A TTL value of no longer than 15 minutes for any wildcard records in TLD zones is suggested.

3 Response Server Configuration Considerations

A response server should be configured to return an indication that the provided services were reached as a result of wildcard processing when the server returns a response to connection requests sent by end user applications. The application protocols responsible

for these connection requests fall into two broad categories: (1) protocols that make a single connection attempt and no further connection attempts if the first attempt fails, and (2) protocols that make multiple connection attempts over time until an attempt succeeds. In both cases, the response server should generate a response that provides meaningful information to the user and the user application to avoid confusion.

G1.2. A response server should be configured to respond to connection attempts on both Transmission Control Protocol (TCP) [3] and User Datagram Protocol (UDP) [4] ports assigned to various application protocols. The number of TCP and UDP ports that must be managed on a response server is large (but finite), with a small number of application protocols expected to produce a majority of the connection attempts.

G1.3. A response server operator should consider the application protocols seen most commonly by their server, and make decisions about the appropriate type of response when a connection attempt is made. In some cases, the most appropriate response might be information intended for presentation to the user for further action. In other cases, the most appropriate response might be a low-level protocol error to discourage additional connection attempts.

3.1 Protocol Separation

A response server operator should make informed decisions about the support provided for each transport and application protocol.

G1.4. The server operator must decide which application protocols will be supported, and must provide appropriate application protocol responses when connection attempts are made by those protocols. Likewise, the server operator must provide appropriate application or network protocol error responses to reject connection attempts made by unsupported application protocols. It may be helpful to include information specific to the actual query name if the query name is sent to the response server by the application protocol.

G1.5. The type and form of the protocol response is best determined in the context of the protocols themselves. A protocol used by a large number of human users with extensive user-initiated, visual characteristics might be best addressed by returning a human-readable response that clearly indicates why a response has been provided by the response server and gives the user options for appropriate next steps.

G1.6. A protocol used by a large number of users with extensive automated operational characteristics might be best addressed by returning an error response

that can be interpreted by a machine or protocol implementation software. Protocols with little user visibility might be best addressed by rejecting the connection attempt outright.

3.2 Application Protocol Responses

Section 3.1 describes general protocol response guidelines. Here we provide specific guidelines for the Hypertext Transfer Protocol (HTTP) [5] and the Simple Mail Transfer Protocol (SMTP) [6], two protocols that are widely used today. Appropriate responses for other protocols can be developed using the guidelines described in Section 3.1.

3.2.1 Hypertext Transfer Protocol

HTTP is an example of a protocol used by a large number of humans with extensive user-initiated, visual characteristics.

G1.7. An HTTP response server should provide a web page with localized human-readable text to clearly explain that the page is being displayed as a result of wildcard processes along with additional information that might be useful to the user. The user can then make informed decisions about their options for subsequent actions.

Web pages produced as a result of wildcard processes can be archived by World Wide Web robots, including those used by many search engines to scan web sites. Given the almost infinite number of domain names that can produce wildcard response web pages, archiving these pages has the potential to include information in search engine databases that would not be found otherwise.

G1.8. An HTTP response server should instruct World Wide Web robots to not access or archive any of the web pages found on the server.

Two methods are widely used today to inform robots of web site access restrictions. The Robots Exclusion Protocol [7] describes a method to inform robots that a site should not be accessed by placing information in a file named *robots.txt*. A *robots.txt* file that instructs a robot to not access any part of a site would contain the following entries:

```
User-agent: *
Disallow: /
```

The World Wide Web Consortium (W3C) HTML 4.01 specification [8] Appendix B describes an alternative method to instruct robots using an HTML META tag. A META tag to instruct a robot to not access any part of a site looks like this:

```
<META name="ROBOTS" content="NOINDEX, NOFOLLOW">
```

3.2.2 Simple Mail Transfer Protocol

SMTP is an example of a protocol used by a large number of humans with extensive automated operational characteristics. SMTP delivery actions typically take place between servers without user involvement, so there is little or no opportunity to correct domain name errors as they are encountered at the server level. An SMTP server that successfully resolves a domain name will typically attempt to deliver a message to the destination server; if the message is initially undeliverable the sending server will make continued delivery attempts over time (up to some maximum number of attempts).

G1.9. A response server should provide a limited SMTP server that returns an SMTP 550 error response for any destination address to indicate that mail delivery is not possible, with text provided to explain why the message is being rejected. This active error response will allow the sending server to take appropriate action for all undeliverable messages.

3.3 Error Responses

G1.10. Unsupported application protocols should be noted through active return of an appropriate transport protocol error response to discourage further connection attempts.

G1.11. The response server should exhibit standard behavior in response to connection attempts to TCP and UDP ports without corresponding applications listening. For TCP-based protocols, the response server should immediately reset the connection (i.e., in response to a client's initial TCP packet with the SYN flag set, the server should return a TCP packet with the RST flag set). For UDP-based protocols, the response server should immediately respond with an ICMP port unreachable message. These are "hard errors" and should cause properly written applications to cease further connection attempts.

G1.12. Response servers should not return other ICMP messages, such as host or network unreachable, which is interpreted by many TCP/IP implementations as a "soft error" and will not stop connection attempts.

G1.13. Likewise, the response server (or network devices in front of it) should not simply drop packets corresponding to connection attempts from unsupported protocols.

Lacking a definite response, an application and/or the underlying TCP/IP software is likely to continue connection attempts until some timeout occurs.

4 Operational Considerations

The success or failure of a zone administrator's wildcard implementation depends largely on the applications that are creating the DNS query traffic being directed to the TLD name servers. As described in Section 3, a response server should be configured to respond to applications that attempt to contact the server.

G1.14. Each TLD likely has unique characteristics in terms of the applications that will produce DNS queries and attempt response server connections, so an active DNS query and response server port monitoring program should be in place to remain aware of usage trends as they develop and change over time.

G1.15. As usage trends develop, the response server should be reconfigured as necessary to provide appropriate services. An active monitoring program is also recommended to detect and respond to denial of service attacks.

G1.16. Publishing the results of the DNS query and response server port monitoring program in a public place, such as on the TLD zone administrator's web site, will allow the user community to see the benefits, including added user efficiencies, provided by the new service.

G1.17. Zone administrators should provide a means to allow contact from individuals and organizations that experience operational issues with a wildcard response server. Support for community-based monitoring increases the likelihood that the zone administrator will be made aware of operational issues in a timely manner. The means provided should provide a balance between anticipated reporting volume and the administrator's ability to respond.

G1.18. The response server and the services it provides should be visible using a well-known host name and Internet Protocol (IP) address. This will allow applications to recognize DNS responses directing them to the response server, and give them an opportunity to take appropriate actions (such as notifying the user) if the wildcard synthesis is unanticipated.

G1.19. A response server should return information to indicate that the provided services were reached as a result of a wildcard entry. In some cases it may be appropriate to not provide services at all, using appropriate application or network error responses to inform the client that services are unavailable.

G1.20. With or without wildcards, if queries contain a spoofed source address the DNS responses can be used to mount a flooding attack against the machine

associated with the spoofed address. Response server operators should anticipate the possibility of packet flooding "denial of service" attacks. TLD name servers receive many queries for nonexistent domain names as a result of application errors and user data entry errors. Consistent with all packet flooding "denial of service" attacks, if the TLD name server is flooded with queries for non-existent domain names, the DNS wildcard responses returned for these queries may result in applications attempting to contact the response server.

5 Conclusions

DNS wildcards have been available to zone administrators since the birth of the DNS in the early 1980s. They are flexible enough to accommodate needs created by the growth of the World Wide Web and increased individual access to the Internet. For example, navigating the World Wide Web can be a frustrating experience when applications present cryptic DNS error messages to human users when the user encounters a nonexistent domain name through data entry, network, or operational errors. A wildcard in the appropriate top-level domain zone can be used to provide the user with information that makes it possible to locate whatever resource the user was searching for in the first place.

6 References

1. Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.
2. Eastlake, D., "Domain Name System Security Extensions" RFC 2535 March 1999.
3. Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.
4. Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.
5. Fielding, R. et al., "Hypertext Transfer Protocol -- HTTP/1.1" RFC 2616, June 1999.
6. Klensin, J. (Editor), "Simple Mail Transfer Protocol" RFC 2821, April 2001.
7. Koster, M., "A Standard for Robot Exclusion", http://www.robotstxt.org/wc/norobots.html, June 1994.
8. Raggett, D. et al.: "HTML 4.01 Specification", http://www.w3.org/TR/html401/, December 1999.