Network Working Group
Internet-Draft
Intended status: BCP
Expires: April 30, 2015

V. Dukhovni

N. Williams, Ed.
Cryptonector

October 27, 2014

Using Wildcard A and AAAA Resource Records in the DNS for Per-User Host-Based Services

draft-dukhovni-using-wildcard-a-rrsets-01

Abstract

This document describes how the use of wildcard A and AAAA resource records (RRs) in the Domain Name System (DNS), optionally coupled with self-service key management for host names that match the wildcards, to create per-user services. This memo describes what should be a best current practice.

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 30, 2015.

Copyright Notice

Table of Contents

1. Wildcard A/AAAA RRs in DNS for Per-User Host-Based Services (PUHBSs)

Often users need to run services (often on multi-user systems) that need host-based service principal names. We describe a method for arranging this without having to share sensitive cryptographic host credentials with users:

1. Publish in the DNS [RFC1034] [RFC1035] a wildcard A and/or AAAA RRset for every hostname on which self-service per-user services will be permitted.

   This means that for any host named, say, "foo.bar.example", one would publish "*.foo.bar.example.", with the same A and/or AAAA RRset as "foo.bar.example.".

2. Provision users with credentials for host-based service names on hostnames of the form: <username>.<hostname.fqdn>.

This allows users to publish "<username>.<hostname-FQDN>" as their services' hostname.

For the rest of this document we shorten "per-user host-based service principal" to "PUHBSP".

And that's it. The difficult part, of course, is (2), but it is possible to adjust existing provisioning systems and/or build new ones to address this. See Section 3

2. Conventions used in this document

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 [RFC2119].

3. Provisioning of Service Credentials, or Self-Service Key Management

Existing standard and non-standard Kerberos [RFC4120] administration and PKIX [RFC5280] online certification authority (CA) protocols may be used for self-service key management of PUHBSP credentials with minimal changes. The main change that is needed is an authorization change on the server-side.

Example protocols whose services may be modified to suit this purpose:

- The "Microsoft Windows 2000 Kerberos Change Password and Set Password Protocols" [RFC3244] and its non-standard predecessors.
- The various other non-standard Kerberos administration protocols that provide interfaces for creating principals and setting their credentials. For example:
  - <http://oskt.secure-endpoints.com/krb5_admin.html>
  - <http://www.eyrie.org/~eagle/software/wallet/>
  - and others.
- kx509 [RFC6717] (a kerberized online CA).
- Any protocol using PKCS#10 [RFC2986].

The principle is general, applying to any authentication mechanism that can authenticate host-based services, not just Kerberos or PKIX.

3.1. Requirements and Recommendations

Three different sorts of authorization decisions are involved:

1. DNS zone administrators decide which hosts get wildcard A/AAAA RRsets.

   DNS zone administrators MAY safely publish wildcard A and AAAA RRsets for all hostnames in their zones, but they may also keep a white-list of such hostnames.

2. The PUHBSP's credential issuer MUST decide whether to issue credentials for any given sub-domain of a given hostname.

   This decision MUST be based on and constrained by the requestor's credentials. For example, the issuer might require that a client authenticate with a user and a host-based service credential or that a user have opted-in any given host whose credentials will be used to acquire the PUHBSP's credentials, it might require that the username label of the PUHBSP correspond to an existing user, and so on.

   Credential issuers SHOULD NOT issue PUHBSP credentials for hostnames with more than one sub-domain label of the actual host's hostname.

3. Hosts themselves SHOULD authorize local requests by local users/processes for credentials for any given sub-domain of the host's hostname.

For example, a host might authenticate local users using traditional operating system facilities (e.g., processes' credentials), then decide whether the local user gets to have a requested sub-domain of the host's hostname.

3.2. Sample Use Case: per-User Web Services

In our deployment it is trivial for users to run their own web services:

- pick an available port number,
- locally request credentials (server certificates and/or Kerberos keys) for running a web service on "<local-username>.<hostname-fqdn>:<port>" (for example, https://jdoe.someserver.foo.example:3000/),
- configure the web service to start on the chosen port number and with the given credentials,
- then start the service.

In this use case no credential acquisition protocols were modified. Instead their services were modified to permit clients with host-based service credentials to acquire credentials for PUHBSPs whose hostname component is a sub-domain of the actual host's.

4. Security Considerations

Some hostnames may be meaningful (e.g., "irc.foo.bar.example" might be taken to mean that an IRC [RFC2812] server is located at that hostname). Users may need to be educated as to how to determine that a fully-qualified hostname is a per-user one.

Credential issuers SHOULD keep white-lists of users and hostnames that are permitted to have PUHBSPs, though not necessarily white-lists of {username, hostname} that are permitted to have PUHBSPs.

Self-service key management services for PKIX [RFC5280] SHOULD use an intermediate, online certification authority (CA), rather than a top-level CA, so as to make it possible to revoke the online CA's certificate. (This is good advice for any CA anyways.)

Note that this scheme does not support users from more than one realm having PUHBSPs on the same host.

4.1. Man-in-the-Middle Attacks by Local Users

Note that a large number of port numbers for running services are available to all users on most operating systems. This means that a local user could start a proxy on one port and forward to another port on the same host, where the second port is a service run by a different user. Mutual authentication generally protects against this attack.

4.1.1. Security Considerations for Per-User HTTP Services

Hypertext Transfer Protocol (HTTP) [RFC2616] used with the SPNEGO-based [RFC4178] HTTP authentication method [RFC4559] is quite common in some environments. Negotiate does not make use of session keys to protect HTTP data and metadata. To safeguard against this, per-user HTTP services MUST use Transport Layer Security (TLS) [RFC5246], not just SPNEGO.

5. IANA Considerations

There are no IANA considerations in this document.

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

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC2812] Kalt, C., "Internet Relay Chat: Client Protocol", RFC 2812, April 2000.

[RFC2986] Nystrom, M. and B. Kaliski, "PKCS #10: Certification Request Syntax Specification Version 1.7", RFC 2986, November 2000.

[RFC3244] Swift, M., Trostle, J., and J. Brezak, "Microsoft Windows 2000 Kerberos Change Password and Set Password Protocols", RFC 3244, February 2002.

[RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005.

[RFC4178] Zhu, L., Leach, P., Jaganathan, K., and W. Ingersoll, "The Simple and Protected Generic Security Service Application Program Interface (GSS-API) Negotiation Mechanism", RFC 4178, October 2005.

[RFC4559] Jaganathan, K., Zhu, L., and J. Brezak, "SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows", RFC 4559, June 2006.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC6717] Hotz, H. and R. Allbery, "kx509 Kerberized Certificate Issuance Protocol in Use in 2012", RFC 6717, August 2012.

Authors' Addresses

Viktor Dukhovni

Email: ietf-dane@dukhovni.org

Nicolas Williams (editor)
Cryptonector, LLC

Email: nico@cryptonector.com