

RADIUS Extensions Working Group
Internet-Draft
Intended status: Experimental
Expires: January 10, 2012

S. Winter
RESTENA
M. McCauley
OSC
July 09, 2011

**NAI-based Dynamic Peer Discovery for RADIUS/TLS and RADIUS/DTLS
draft-ietf-radext-dynamic-discovery-03**

Abstract

This document specifies a means to find authoritative RADIUS servers for a given realm. It can be used in conjunction with RADIUS/TLS and RADIUS/DTLS.

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 January 10, 2012.

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.	Requirements Language	3
1.2.	Terminology	3
2.	DNS-based NAPTR/SRV Peer Discovery	3
2.1.	Applicability	3
2.2.	DNS RR definition	3
2.3.	Realm to AAA server resolution algorithm	5
3.	Security Considerations	8
4.	IANA Considerations	8
5.	Normative References	8

1. Introduction

1.1. Requirements Language

In this document, several words are used to signify the requirements of the specification. 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](#)]

1.2. Terminology

RADIUS/TLS Client: a RADIUS/TLS [[I-D.ietf-radext-radsec](#)] instance which initiates a new connection.

RADIUS/TLS Server: a RADIUS/TLS [[I-D.ietf-radext-radsec](#)] instance which listens on a RADIUS/TLS port and accepts new connections

RADIUS/TLS node: a RADIUS/TLS client or server

2. DNS-based NAPTR/SRV Peer Discovery

2.1. Applicability

Dynamic server discovery as defined in this document is only applicable for AAA transactions where a RADIUS server receives a request with a realm for which no home RADIUS server is known. I.e. where static server configuration does not contain a known home authentication server, or where the server configuration explicitly states that the realm destination is to be looked up dynamically. Furthermore, it is only applicable for new user sessions, i.e. for the initial Access-Request. Subsequent messages concerning this session, for example Access-Challenges and Access-Accepts use the previously-established communication channel between client and server.

2.2. DNS RR definition

DNS definitions of RADIUS/TLS servers can be either S-NAPTR records (see [[RFC3958](#)]) or SRV records. When both are defined, the resolution algorithm prefers S-NAPTR results (see section [Section 2.3](#) below).

This specification defines three S-NAPTR service tags: "aaa+auth", "aaa+acct" and "aaa+dynauth". This specification defines two S-NAPTR protocol tags: "radius.tls" for RADIUS/TLS [[I-D.ietf-radext-radsec](#)] and "radius.dtls" for RADIUS/DTLS [[I-D.dekok-radext-dtls](#)].

This specification defines the SRV prefix "_radius.tls.tcp" for RADIUS over TLS [[I-D.ietf-radext-radsec](#)] and "_radius.tls.udp" for RADIUS over DTLS [[I-D.dekok-radext-dtls](#)]. It is expected that in most cases, the label used for the records is the DNS representation (punycode) of the literal realm name for which the server is the AAA server.

However, arbitrary other labels may be used if, for example, a roaming consortium uses realm names which are not associated to DNS names or special-purpose consortia where a globally valid discovery is not a use case. Such other labels require a consortium-wide agreement about the transformation from realm name to lookup label.

Examples:

- a. A general-purpose AAA server for realm example.com might have DNS entries as follows:

```
example.com. IN NAPTR 50 50 "s" "aaa+auth:radius.tls" ""
_radius.tls.tcp.foobar.example.com.

_radius.tls.tcp.foobar.example.com. IN SRV 0 10 2083
radsec.example.com.
```

- b. The consortium "foo" provides roaming services for its members only. The realms used are of the form enterprise-name.example. The consortium operates a special purpose DNS server for the (private) TLD "example" which all AAA servers use to resolve realm names. "Bad, Inc." is part of the consortium. On the consortium's DNS server, realm bad.example might have the following DNS entries:

```
bad.example IN NAPTR 50 50 "a" "aaa+auth:radius.dtls" ""
"very.bad.example"
```

- c. The eduroam consortium uses realms based on DNS, but provides its services to a closed community only. However, a AAA domain participating in eduroam may also want to expose AAA services to other, general-purpose, applications (on the same or other AAA servers). Due to that, the eduroam consortium uses the service tag "x-eduroam" for authentication purposes and eduroam AAA servers use this tag to look up other eduroam servers. An eduroam participant example.org which also provides general-purpose AAA on a different server uses the general "aaa+auth" tag:

```
example.org. IN NAPTR 50 50 "s" "x-eduroam:radius.tls" ""
_radius.tls.tcp.eduroam.example.org.
```



```
example.org. IN NAPTR 50 50 "s" "aaa+auth:radius.tls" ""
_radiustls._tcp.aaa.example.org

_radiustls._tcp.eduroam.example.org. IN SRV 0 10 2083 aaa-
eduroam.example.org.

_radiustls._tcp.aaa.example.org. IN SRV 0 10 2083 aaa-
default.example.org.
```

2.3. Realm to AAA server resolution algorithm

Input I to the algorithm is the RADIUS User-Name attribute with content of the form "user@realm"; the literal @ sign being the separator between a local user identifier within a realm and its realm. The use of multiple literal @ signs in a User-Name is strongly discouraged; but if present, the last @ sign is to be considered the separator. All previous instances of the @ sign are to be considered part of the local user identifier. Output O of the algorithm is a set of hostname:port and an associated order/preference; the set can be empty.

Note well: The attribute User-Name is defined to contain UTF-8 text. In practice, the content may or may not be UTF-8. Even if UTF-8, it may or may not map to a domain name in the realm part. Implementors MUST take possible conversion error paths into consideration when parsing incoming User-Name attributes. This document describes server discovery only for well-formed realms mapping to DNS domain names in UTF-8 encoding. The result of all other possible contents of User-Name is unspecified; this includes, but is not limited to:

Usage of separators other than @

Usage of multiple @ separators

Encoding of User-Name in local encodings

UTF-8 realms which fail the conversion rules as per [[RFC5891](#)]

The algorithm to determine the RADIUS server to contact is as follows:

1. Determine P = (position of last "@" character) in I.
2. generate R = (substring from P+1 to end of I)
3. Optional: modify R according to agreed consortium procedures

4. Using the host's name resolution library, perform a NAPTR query for R. The name resolution library may need to convert R to a different representation, depending on the resolution backend used. If no result, continue at step 9. If name resolution returns with error, $O = \{ \}$. Terminate.
5. Extract NAPTR records with service tag "aaa+auth", "aaa+acct", "aaa+dynauth" as appropriate.
6. If no result, continue at step 9.
7. Evaluate NAPTR result(s) for desired protocol tag, perform subsequent lookup steps until lookup yields one or more hostnames. $O = (\text{set of } \{\text{Order/Preference, hostname:port}\} \text{ for all lookup results})$.
8. Terminate.
9. Generate $R' = (\text{prefix R with } _radiustls._tcp. \text{ or } _radiustls._udp)$
10. Using the host's name resolution library, perform SRV lookup with R' as label.
11. If name resolution returns with error, $O = \{ \}$. Terminate.
12. If no result, $O = \{ \}$; terminate.
13. Perform subsequent lookup steps until lookup yields one or more hostnames. $O = (\text{set of } \{\text{Order/Preference, hostname}\} \text{ for all hostnames})$. Terminate.

Example: Assume a user from the Technical University of Munich, Germany, has a RADIUS User-Name of "foobar@tu-m[U+00FC]nchen.example". The name resolution library on the RADIUS client uses DNS for name resolution. If DNS contains the following records:

```
xn--tu-mnchen-t9a.example. IN NAPTR 50 50 "s" "aaa+
auth:radius.tls" "" _radiustls._tcp.xn--tu-mnchen-t9a.example.
```

```
xn--tu-mnchen-t9a.example. IN NAPTR 50 50 "s" "fooservice:
bar.dccp" "" _abc._def.xn--tu-mnchen-t9a.example.
```

```
_radiustls._tcp.xn--tu-mnchen-t9a.example. IN SRV 0 10 2083
radsec.xn--tu-mnchen-t9a.example.
```



```
_radiustls._tcp.xn--tu-mnchen-t9a.example.  IN SRV 0 20 2083  
backup.xn--tu-mnchen-t9a.example.
```

```
radsec.xn--tu-mnchen-t9a.example.  IN AAAA 2001:0DB8::202:44ff:  
fe0a:f704
```

```
radsec.xn--tu-mnchen-t9a.example.  IN A 192.0.2.3
```

```
backup.xn--tu-mnchen-t9a.example.  IN A 192.0.2.7
```

Then the algorithm executes as follows, with I =
"foobar@tu-m[U+00FC]nchen.example", and no consortium name mangling
in use:

1. P = 7
2. R = "tu-m[U+00FC]nchen.example"
3. NOOP
4. [name resolution library converts R to xn--tu-mnchen-t9a.example] Query result: (50 50 "s" "aaa+auth:radius.tls" ""
_radiustls._tcp.xn--tu-mnchen-t9a.example. ; 50 50 "s"
"fooservice:bar.dccp" "" _abc._def.xn--tu-mnchen-t9a.example.)
5. Result: 50 50 "s" "aaa+auth:radius.tls" "" _radiustls._tcp.xn--
tu-mnchen-t9a.example.
6. NOOP
7. O = {(10,radsec.xn--tu-mnchen-t9a.example.:2083),(20,backup.xn--
tu-mnchen-t9a. example.:2083)}
8. Terminate.
9. (not executed)
10. (not executed)
11. (not executed)
12. (not executed)
13. (not executed)

The implementation will then attempt to connect to two servers, with
preference to radsec.xn--tu-mnchen-t9a.example.:2083, using either
the AAAA or A addresses depending on the host configuration and its

IP stack's capabilities.

3. Security Considerations

When using DNS without DNSSEC security extensions, the replies to NAPTR, SRV and A/AAAA requests as described in section [Section 2](#) can not be trusted. RADIUS transports have an out-of-DNS-band means to verify that the discovery attempt led to the intended target: certificate verification or TLS-PSK keys.

4. IANA Considerations

This document requests IANA registration of the following S-NAPTR parameter:

- o Application Service Tags
 - * aaa+auth
 - * aaa+acct
 - * aaa+dynauth
- o Application Protocol Tags
 - * radius.tls
 - * radius.dtls

5. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3958] Daigle, L. and A. Newton, "Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)", [RFC 3958](#), January 2005.
- [RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", [RFC 5891](#), August 2010.
- [I-D.dekok-radext-dtls] DeKok, A., "DTLS as a Transport Layer for RADIUS", [draft-dekok-radext-dtls-03](#) (work in progress), July 2010.

[I-D.ietf-radext-radsec] Winter, S., McCauley, M., Venaas, S., and K. Wierenga, "TLS encryption for RADIUS", [draft-ietf-radext-radsec-09](#) (work in progress), July 2011.

Authors' Addresses

Stefan Winter
Fondation RESTENA
6, rue Richard Coudenhove-Kalergi
Luxembourg 1359
LUXEMBOURG

Phone: +352 424409 1
Fax: +352 422473
EMail: stefan.winter@restena.lu
URI: <http://www.restena.lu>.

Mike McCauley
Open Systems Consultants
9 Bulbul Place
Currumbin Waters QLD 4223
AUSTRALIA

Phone: +61 7 5598 7474
Fax: +61 7 5598 7070
EMail: mikem@open.com.au
URI: <http://www.open.com.au>.

