RADIUS Extensions Working Group S. Winter
Internet-Draft RESTENA
Intended status: Experimental M. McCauley
Expires: January 14, 2010 OSC
July 13, 2009

NAI-based Dynamic Peer Discovery for RADIUS over TLS and DTLS
draft-ietf-radext-dynamic-discovery-01

Status of This Memo

Copyright Notice

Abstract

This document specifies a means to find authoritative AAA servers for a given NAI realm as defined in [RFC4282]. It can be used in

conjunction with RADIUS over TLS and RADIUS over DTLS.

Table of Contents

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

RadSec node: a RadSec client or server

RadSec Client: a RadSec instance which initiates a new connection.

RadSec Server: a RadSec instance which listens on a RadSec port and accepts new connections

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 AAA server receives a request with a NAI realm for which no home AAA 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, Access-Accepts, Accounting Messages or Change-of-Authorisation messages use the previously-established communication channel between client and server.

2.2. DNS RR definition

DNS definitions of RadSec servers can be either NAPTR records or SRV records. When both are defined, the resolution algorithm prefers NAPTR results (see section Section 2.3 below). The NAPTR service field used is "AAAS+RADSECT". The SRV prefix used is "_radsec._tcp". 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" "AAAS+RADSECT" ""
_radsec._tcp.foobar.example.com.

_radsec._tcp.example.com.  IN SRV 0 10 2083
radsec.example.com.
```

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

```
bank-rupt.foobankroam IN NAPTR 50 50 "a" "AAAS+RADSECT" ""
"triple-a.bank-rupt.com"

_radsec._tcp.bank-rupt.foobankroam IN SRV 0 10 2083 triple-a-
backup.bank-rupt.com"
```

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 labels prefixed with "eduroam." and eduroam AAA servers use these labels to look up servers. An eduroam participant which also provides general-purpose AAA on a different server might have the following DNS entries:

```
eduroam.restena.lu.  IN NAPTR 50 50 "a" "AAAS+RADSECT" "" aaa-
eduroam.restena.lu

restena.lu.  IN NAPTR 50 50 "a" "AAAS+RADSECT" "" aaa-
default.restena.lu

_radsec._tcp.eduroam.restena.lu.  IN SRV 0 10 2083 aaa-
eduroam.restena.lu.
```

_radsec._tcp.restena.lu. IN SRV 0 10 2083 aaa-default.restena.lu.

2.3. Realm to AAA server resolution algorithm

Input I to the algorithm is a User-Name in the form of a NAI as defined in [RFC4282] as extracted from the User-Name attribute in an Access-Request. Output O of the algorithm is a set of hostname:port and an assoiciated order/preference; the set can be empty.

Note well: The attribute User-Name does not necessarily contain well-formed NAIs and may not even contain well-formed UTF-8 strings. This document describes server discovery only for well-formed NAIs 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

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

1. Determine P = (position of first "@" 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 service "AAAS+RADSECT" for R

5. If name resolution returns with error, O = { }. Terminate.

6. If no result, continue at step 9.

7. Evaluate NAPTR result, perform subsequent lookup steps until lookup yields one or more hostnames. O = (set of {Order/Preference, hostname:port} for all lookup results).

8. Terminate.

9. Generate R' = (prefix R with "_radsec._tcp.")

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 = (set of {Order/Preference, hostname} 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". If DNS contains the following records:

```
xn--tu-mnchen-t9a.example.  IN NAPTR 50 50 "s" "AAAS+RADSECT" ""
_radsec._tcp.xn--tu-mnchen-t9a.example.

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

_radsec._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. Query result: (0 10 2083 radsec.xn--tu-mnchen-t9a.example. ; 0 20 2083 backup.xn--tu-mnchen-t9a.example.)

5. NOOP

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 security, 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 (TLS/DTLS: ceritifcate verification or TLS shared secret ciphers; UDP/TCP: the RADIUS shared secret) and are safe from DNS-based redirection attacks. [Note: assuming here that a hypothetical RADIUS/UDP SRV discovery will NOT deliver the shared secret in the DNS response!]

The discovery process is always susceptible to bidding down attacks if a realm has SRV records for RADIUS/UDP and/or RADIUS/TCP as well as for RADIUS/TLS and/or RADIUS/DTLS. While the SRV query will expose both transports, an attacker in the routing path might suppress the subsequent A/AAAA results for the TLS or DTLS peer and trick the initiating peer into using the weakly protected UDP or TCP transports. The use of DNSSEC can not fully mitigate this attack, since it does not provide a means to detect packet suppression. The only way to disable such bidding down attacks is by intiating connections only to the peer(s) which match or exceed a configured minimum security level. All implementations SHOULD provide a means to configure the administratively desired minimum security level.

4. IANA Considerations

This document contains no actions for IANA. Maybe. Not sure about the labels "AAAS+RADSECT" and "_radsec._tcp.".

5. Normative References

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

[RFC4282] Aboba, B., Beadles, M., Arkko, J., and P. Eronen, "The Network Access Identifier", RFC 4282, December 2005.

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.