

Kerberos Realm Descriptors in DNS (KREALM)
draft-vanrein-dnstxt-krb1-03

Abstract

This specification defines methods to determine Kerberos realm descriptive information for services that are known by their DNS name. Currently, finding such information is done through static mappings or educated guessing. DNS can make this process more dynamic, provided that DNSSEC is used to ensure authenticity of resource records.

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 March 6, 2016.

Copyright Notice

Copyright (c) 2015 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	2
2.	The KREALM Resource Record	3
3.	Defining Home Records and Reference Records	4
4.	Querying Kerberos Realm Descriptors	6
4.1.	Querying a Domain's Kerberos Realm Descriptors	7
4.2.	Querying a Host's Kerberos Realm Descriptors	7
5.	Publishing Kerberos Realm Descriptors	8
6.	Tags in Kerberos Realm Descriptors	9
6.1.	Kerberos Realm Descriptor Tag "realm"	9
6.2.	Kerberos Realm Descriptor Tag "service"	10
6.3.	Kerberos Realm Descriptor Tag "admin"	12
7.	Efficiency Considerations	13
8.	Privacy Considerations	14
9.	Security Considerations	14
10.	IANA Considerations	16
11.	References	16
11.1.	Normative References	17
11.2.	Informative References	17
	Author's Address	18

[1. Introduction](#)

When a Kerberos client contacts a service, it needs to obtain a service ticket, and for that it needs to contact the KDC for a realm under which the service is run. To map a service name into a realm name and then into a KDC, clients tend to use static mappings or educated guesses; the client's KDC may or may not be involved in this process. Through DNS, such mappings could be dynamically expanded, permitting more flexibility than under the current practice.

Two mappings are needed for the given scenario. One is a mapping from the FQDN of a service to its realm name; the other is a mapping from the realm name to the Kerberos-specific services such as the KDC. The latter mapping is published in SRV records [[RFC4120](#)] and such traffic is protected by the Kerberos protocol itself. The first mapping however, has hitherto not been standardised and is ill-advised over unsecured DNS because the published information is then neither validated by DNS nor does it lead to a protocol that could provide end-to-end validation for it.

With the recent uprise of DNSSEC, it is now possible to make a reliable judgement on the authenticity of data in DNS, which enables

the standardisation of the first mapping in the form of resource records in secured DNS.

This specification defines how to publish and process Kerberos Realm Descriptors using a newly defined resource record type KREALM. Each of these records holds a series of tagged string values. A few of these are defined below, others may be added by future specifications.

2. The KREALM Resource Record

This specification introduces a new DNS resource record that serves as a Kerberos Realm Descriptor. The name for this DNS resource record type is KREALM, and its numeric value is TBD1. The corresponding RDATA format is as follows:

```
+---+---+---+---+---+---+---+---+---+---+---+---+
/                               KREALM-DATA                               /
+---+---+---+---+---+---+---+---+---+---+---+---+
```

The syntax of KREALM-DATA is the DER encoding of the following ASN.1 syntax:

```
KREALM-DATA ::= SEQUENCE {
    versionNumber INTEGER (0..) DEFAULT 0,
    SET OF SEQUENCE {
        tag IA5String,
        value UTF8String
    }
}
```

This specification makes the assumption that all future tag definitions will define their corresponding value fields to be represented in strings.

DISCUSS: [RFC 4120](#) defines realm names with the KerberosString type which is a GeneralString, but it then advises to constrain it to IA5String or else risk interoperability problems. It is worth noting that the ESC "%" "G" prefix defined in ISO 2022 can be used to introduce an UTF8String in the KerberosString, and that implementations even exist that insert UTF8String in KerberosString fields without even that escape. It's a jungle out there! Defining UTF-8 for this new field type does not seem to be such a stretch; it includes the IA5String subset and it also keeps the door ajar for future attempts at I18N of realm names and other Kerberos parameters. OTOH, OCTETSTRING would probably be too general, and place too much faith in wire representations to be suitable for comparison.

The advised format for the resource data in master zone files is the base64-encoded KREALM-DATA content in DER-encoding. Although most data will be printable string data, it need not adhere to the ASCII character set or another constrained set the DNS processing software may assume, it may include NUL bytes and have a somewhat complex grammar; all these aspects would complicate master zone files and thus make them more error-prone; hence the representation in base64 format. The base64-represented data may be interspersed with white space, including even line breaks if the usual DNS zone file notation with parenthesis is used.

Tags are registered with IANA, and this document defines the first few. A special range of tags starting with "x-" is available for local or experimental use. Implementations MAY safely ignore tags (and the corresponding value) that are not known to them. An application that does not recognise a tag name MUST silently discard it.

Software implementing this specification can be recognised by versionNumber 0; this is also the default versionNumber when absent from the encoding. Future updates to this specification MUST use another versionNumber IF they invalidate any assumptions made in this specification. Such new applications SHOULD advise how to setup DNS in a backward-compatible manner; they might for instance publish both old and new styles of KREALM records.

Clients requesting KREALM records MUST ensure that the record uses proper syntax, including the string formats specified for tag and value fields and a versionNumber that the client understands.

Multiple KREALM records may be supplied under a queried name, and there may be multiple that adhere to this syntax; these present alternatives that can be tried, possibly with different versionNumber values. Since DNS does not supply them in any order, the DNS client can choose freely in what order to process these records. One possibility is a DNS client that prefer certain KREALM records over others.

In this general form, there are no constraints on the number of permissible occurrences of a tag in one or more KREALM records, but tags MUST define whether multiple occurrences are permitted, and if so, what their interpretation is.

3. Defining Home Records and Reference Records

One of the tags defined by this specification is the "realm" tag [[Section 6.1](#)] that specifies a realm name. Domain-style realm names can be mapped [[Section 7.2.3 of \[RFC4120\]](#)] to DNS names. Those realm

names in "realm" tags of KREALM records that map to the FQDN at which that KREALM record was found, are hereby defined as "Home Realms".

The realm names specified in "realm" tags need not all be domain-style realm names [[Section 6.1 of \[RFC4120\]](#)]; however, since a translation to DNS is used in the definition of a Home Realm, only domain-style realm names can ever be Home Realms. In practice, since the only impact of mapping between domain-style realm names to DNS names is that DNS names are case-insensitive, a Home Realm is a case-insensitive equivalent of the DNS name at which its containing KREALM record is published.

KREALM records of which all "realm" tags are Home Realms, are hereby defined as "Home Records". Note how KREALM records are not defined to be Home Realms when they contain a mixture of Home Tags and "realm" tags that are not Home Tags. Note how a single DNS name may hold many KREALM records, some of which are Home Records and some of which are not. Given the aforementioned properties of case sensitivity, all "realm" tags in a Home Realm will be case-insensitive equivalents of the DNS name at which they are published.

KREALM records that are not Home Records may be referred to as Reference Records, since their function is to reference a realm, whose name may be translated to a DNS name that may in turn provide a Home Record. Insofar as a KREALM record is located through a Reference Record, secure processing of the referenced KREALM record would require verification that it is a Home Record. The term Reference Record will still be used when other than domain-style realm names are used, so when it may be impossible to locate a Home Record.

KREALM records SHOULD NOT combine "realm" tags that are Home Tags with others that are not; it is considered bad style and blocks processing. When such KREALM are encountered they MUST be ignored.

In addition to referencing a Home Record, which is in fact an optional part of a realm's configuration in DNS, there is the additional possibility that a Reference Record for a domain-styled realm name points to a DNS name that holds information about the location of a KDC [[Section 7.2.3 of \[RFC4120\]](#)] and other realm-specific services. Such services must incorporate their own validation mechanisms to be secure, since the Reference Record cannot be trusted to serve the interest of the referenced realm; at best it may be trusted to serve the interest of the referring DNS name that holds the Reference Record. Note however that the customary DNS definitions such as for a KDC already employ Kerberos mechanisms (namely, knowledge of a password or holding a ticket) to perform such

trust validation; this is why they could be introduced in unsecured DNS in the past.

The following specifications are strict about their reliance on DNSSEC for KREALM records. The reason for this is to ensure that both the pointer to the actual service and its containing realm are inserted into DNS by the same responsible party. When combined with cryptographic assurance of reaching the proper KDC for a realm, this provides a secure mechanism through which realms can be contacted, regardless of whether they are the realm into which a user logged on.

Mechanisms for cryptographic assurance of reaching realms is standard in Kerberos implementations; based on DNSSEC this may even be extended with more dynamic mechanisms, but that is not defined in this specification. A result that this specification may have is that the user knows the realms to ask for, even if it is a realm that was hitherto unknown. In that sense, the KREALM record can be a stepping stone in loosely connected links between Kerberos realms.

4. Querying Kerberos Realm Descriptors

The following subsections define two procedures for finding Kerberos realm descriptors for the DNS name of a service. One procedure starts from a domain name, the other starts from the host name of a service.

When dealing with services found through DNS SRV [[RFC2782](#)], a choice between the use of a domain name or host name is possible. In these situations, the FQDN of the SRV queries, without the `_Service._Proto` prefix, MUST be used in the procedure for domain name queries, and the procedure for querying a domain should be followed rather than the procedure for a host name.

Since DNS in general cannot be considered secure, the client MUST dismiss any DNS responses that are Insecure, Bogus or Indeterminate [[Section 5 of \[RFC4033\]](#)]. Only the remaining Secure responses are to be taken into account. This specification does not require that the client validates the responses by itself, but a client deployment SHOULD NOT accept DNS responses from a trusted validating DNS resolver over untrusted communication channels.

To give one possible implementation, a Kerberos client may send DNS queries with the Authentic Data (AD) bit set to enable DNSSEC [[Section 5.7 of \[RFC6840\]](#)], and require that the Authenticated Data bit is set in the response to indicate [[RFC3655](#)] the Secure state for answer and authority sections of the response. When the DNS traffic to and from the validating resolver is protected, for instance because that resolver is reached over a loopback interface, then the

Kerberos client has implemented the requirements for Secure use of the answer and authority sections in DNS responses.

The result may contain KREALM records that do not adhere to the syntax of this specification; such KREALM records MUST be removed from the result. In the sequence of tag-and-value pairs, there may be tags that are unknown; such tags and their value MUST be ignored when further processing the results. Finally, some tags are specifically registered with IANA for use in Home Tags, and those MUST be ignored if the KREALM record is not a Home Record.

When no Secure DNS responses are received, this procedure MUST be terminated without extracting realm descriptive information from DNS. Such termination need not be fatal; non-DNS procedures may exist to find a realm name.

4.1. Querying a Domain's Kerberos Realm Descriptors

To find Kerberos Realm Descriptors for a domain name, a DNS client conducts a KREALM query targeted directly at the domain name.

Where this specification speaks of querying a domain, its interpretation of a domain is that of a name space, which may or may not have a host attached, but which is likely to have services attached, for instance through AAAA, MX or SRV records. Domain names also occur in many naming schemes after an optional username and @ symbol, such as the domain name (that also happens to be termed "realm", but without connection to Kerberos realms) in a Network Access Identifier [[RFC4282](#)].

4.2. Querying a Host's Kerberos Realm Descriptors

To find a Kerberos Realm Descriptor for a host name, a KREALM query is performed by upward iteration towards the DNS root, but never risking to cross over to different authoritative control, as marked by a SOA record.

Although upward iteration through the DNS could be detrimental to DNS performance in pathological cases, most practical zones will not have many levels of host names, and as a result the search for the KREALM records should not take long. Also, as described in [Section 5](#) below, operators do have the ability to define additional KREALM records at strategic names.

The algorithm starts by setting the current name to the host name. Then, it goes through the following loop:

1. A KREALM query is launched against the current name.

2. When the query times out in spite of resending or attempting other name servers, then the algorithm ends in failure.
3. When a Secure DNS response arrives holding one or more KREALM records, then these can be processed; the algorithm ends successfully.
4. When a Secure denial is received in the form of an NSEC [[RFC3845](#)] or NSEC3 [[RFC5155](#)] record, the algorithm continues.
5. When the type bit map in the secure denial indicates the presence of a SOA record under the current name, then no further iterations are possible, and the algorithm ends in failure.
6. The first label is removed from the current name, and the loop repeats.

A failure from the above algorithm indicates that no Kerberos Realm Descriptor records could be found, and so that no assumptions may be made on those.

Before moving up in the DNS hierarchy, a resolver MUST ensure the absence of a SOA record for the DNS name that was unsuccessfully queried for a KREALM record; the algorithm as shown does this efficiently, based on the assumption that the type bit map from the secure denial is available to the resolver. Resolvers MAY however follow an alternative strategy, namely assuring the absence of SOA records by making explicit queries for it.

[5.](#) Publishing Kerberos Realm Descriptors

KREALM records are best published at the DNS-mapped name for a Kerberos realm, where they can be setup as Home Records. In many practical situations, this DNS-mapped realm name will match the apex of a zone. It is also possible to strategically position KREALM records at lower positions in the DNS hierarchy, although those could not be setup as Home Records.

When a KREALM record is published for a certain DNS name, it will cover all MX and SRV records for that DNS name, as well as all host names defined at the same DNS name or hierarchically lower but in the same zone. SRV and MX records for hierarchically lower DNS names need a separate KREALM record.

For child zones that contain service definitions that fall under a parent zone, the KREALM records must be repeated to supporting finding the Kerberos Realm Descriptor for the child zones' services.

This is a result of the refusal of the algorithms to cross-over to parent zones.

The additional mentioning of KREALM records on hierarchically lower names than the DNS-mapped realm name is also useful when the contents of the KREALM records needs to be modified. One form of overriding KREALM definitions worth noting is one that does not define a Kerberos realm at all; such a record is useful to undefine any realm names that are defined higher up the DNS hierarchy.

Note that KREALM records with wildcard names will not work. All host names and most domain names define at least one resource record (of any type) with the name that the wildcard should cover. These defined names cause the wildcards to be suppressed [[RFC4592](#)] from DNS responses, even when querying a non-existent KREALM record.

6. Tags in Kerberos Realm Descriptors

The names of tags are partitioned into two types:

- o Tags that are not Home Tags can be used in any KREALM records;
- o Home Tags MUST NOT be published except in Home Records;
- o Home Tags MUST NOT be processed unless they occur in Home Records.

Any Home Tags that occur in other KREALM records than Home Records MUST be ignored.

6.1. Kerberos Realm Descriptor Tag "realm"

The tag "realm" MAY be present in all KREALM records, and it MUST be recognised and processed by implementations of this specification; in other words, the tag is not optional.

The value of a "realm" tag provides a realm name for the queried FQDN. The permissible values of this tag conform to the permissible names of realm names [[Section 6.1 of \[RFC4120\]](#)], which a conforming application MUST validate before processing the value. This includes, but is not limited to, domain-style realm names. Since the value field is a general UTF8String, it is to be treated as a case-sensitive string [[RFC4343](#)], just like realm names.

It is possible to define zero "realm" tags in a KREALM record. This indicates that no realm is defined by that record. This is not an invalid condition; other KREALM records, if any, or other service-to-realm mappings may still be used.

When multiple "realm" tags occur in one KREALM record, then they present alternative suggestions to combine with all other tags in the same KREALM record.

An example use of the "realm" tag in a TXT record is

```
example.com.  IN KREALM MBgxFjAUFgVyZWfSbQwLRVhBTvBMRS5DT00=
```

Written out, the RDATA holds the following DER representation:

```
SEQUENCE
  SET
    SEQUENCE
      IA5STRING    realm
      UTF8STRING   EXAMPLE.COM
```

Since the value of all "realm" tags map to the FQDN of this KREALM record, this KREALM record is a Home Record. As a result, tags that are registered as Home Tags may be added to this realm descriptor.

In addition to the previous example, the following tag indicates that there is no realm, and so it is useless to request a service ticket for ftp.example.com, as far as this domain descriptor in DNS is concerned:

```
ftp.example.com.  IN KREALM MAixAA==
```

The RDATA for this KREALM record encodes no tags and no values at all:

```
SEQUENCE
  SET
```

6.2. Kerberos Realm Descriptor Tag "service"

The tag "service" MAY be present in any KREALM record, and it SHOULD be recognised by implementations of this specification. Occurrence of the tag is optional.

The value of a "service" tag is the name of a service, as used in principal names. This can be used as a hint to clients that need to match "service" tags. The occurrence of a "service" tag and a "realm" tag in the same KREALM record is a hint that a service ticket for the combination probably exists. Note that the value of this tag is a general UTF8String, and that it is case-sensitive [[RFC4343](#)].

The purpose of this tag is to enable clients to locally select alternatives that it may wish to pursue; adding a "service" tag may

improve the speed of resolution when multiple alternatives are listed in DNS, which is especially fruitful when future initiatives would require public key cryptography for realm crossover.

When no "service" tag is defined in a KREALM record, then no hint for selection is available; processing must then continue under the assumption that any desired service name may be available for the realm description. In contrast, when one or more "service" tags are defined in a KREALM record, then this set may be considered a complete specification of available services. Note that multiple KREALM records may exist, each of which may or may not define "service" tags.

When multiple "service" tags occur in one KREALM record, then they present alternative suggestions to combine with all other tags in the same KREALM record.

An example use of this tag in a KREALM record is

```
www.example.com. IN KREALM ( ME8xTTA0FgdzZXJ2aWNlDANmdHAwDxYH
c2Vydm1jZQwESFRUUDAUFGvYyZWFSbQwL RVhBTVMRMS5DT00wFBYFcmlVhbG0MC0VY
QU1QTEUuT1JH )
```

This RDATA contains the following data structure:

```
SEQUENCE
  SET
    SEQUENCE
      IA5STRING    service
      UTF8STRING   ftp
    SEQUENCE
      IA5STRING    service
      UTF8STRING   HTTP
    SEQUENCE
      IA5STRING    realm
      UTF8STRING   EXAMPLE.COM
    SEQUENCE
      IA5STRING    realm
      UTF8STRING   EXAMPLE.ORG
```

This matches the following principal names, found by iterating over all combinations of "server" and "realm" values:

- o HTTP/www.example.com@EXAMPLE.COM
- o HTTP/www.example.com@EXAMPLE.ORG
- o ftp/www.example.com@EXAMPLE.COM

- o ftp/www.example.com@EXAMPLE.ORG

Apart from the case-insensitivity of the DNS name "www.example.com", this list is a complete list of principal names matched by the KREALM record. For instance, a service named "krbtgt" is not described by this KREALM record.

6.3. Kerberos Realm Descriptor Tag "admin"

The tag "admin" MAY be present in KREALM Home Records, and it MAY be recognised by implementations of this specification. Occurrence of the tag is optional.

The value of an "admin" tag is the principal name of an administrator, with or without their Kerberos realm name. This can be used by outside entities to assume realm-administrative rights for certain principals. Specifically, an administrator is supposed to exercise control over both administrators and non-administrators of a realm.

The value of this tag is always a single string, which may occur in a variety of shapes, all known from the customary tools in use for Kerberos. A few examples are:

- o john
- o john/admin
- o john/admin@EXAMPLE.COM
- o john\@example.org@EXAMPLE.COM

The first two forms do not include a realm name, and will combine with "realm" tag values to form the principal name. The third form explicitly mentions that administrator's realm name, which may be foreign to the realms being controlled. The last form is an example of a so-called enterprise principal name [[RFC6806](#)], for which a backslash is used to escape the first @ symbol to avoid its interpretation as a separator between principal name and realm name.

Multiple "admin" tags can be defined; each then defines an alternate administrator, each of which are suggested to be equally capable of controlling principal names under the realm.

When no "admin" tags are defined in a KREALM record, then it simply does not define any administrators for the realms that it describes. Administrators are still likely to exist, but they are simply not published in the KREALM record. In general, it is safe to assume (up

to the trust level of the KREALM publication process) that an existing "admin" tag represents an administrator's principal name, but it is not safe to assume that the absence of an "admin" tag for a given principal name implies that this principal name does not represent an administrator.

In general, every "admin" tag in a KREALM record defines a Kerberos user that can control the principal name for every realm described in the same KREALM record. A realm name mentioned as part of the "admin" tag value may indicate a foreign realm's administrator controlling the principal names of the defined realms.

Since "admin" tags may only be relied upon when mentioned in Home Records, it may be necessary upon encountering another KREALM record to first follow it to the realm name, then translate that to a DNS name, and use that to lookup the Home Record that may contain "admin" tags.

7. Efficiency Considerations

KREALM records are useful to define realm names for servers whose DNS information is not statically mapped in a Kerberos setup. This may simplify operative control of such static mappings. It may also introduce more dynamicity, which may be useful for such things as realm crossover.

Since realm names cannot be derived directly from DNS names, clients tend to construct various principal names by appending all the realm names that they are aware of, and attempting to obtain a service ticket for each in turn, until one is found. The KDC may also perform such actions, and return a reference [[RFC6806](#)] to a realm for consideration. In general, the list of service principal names that may be considered can be relatively long.

The use of "realm" tags help the client to focus on those realms for which a service has a name defined. This limit the list of realm names to attempt to those realm names that the service suggests. The client does not need to guess as heavily. Similarly, the combined use of "service" tags helps to select only those KREALM records that further constrain the scope to search.

Limiting the length of the list of ticket requests is especially useful for situations with realm crossover when this involves public-key cryptography, because such algorithms are much slower than the symmetric algorithms that are normally used for Kerberos.

The combined publication of multiple "realm" tags with multiple "service" tags and/or with multiple "admin" tags enables a compact

representation of variations that a client should iterate over, without the need to store the resulting cartesian product in DNS.

The use of an iterative procedure that moves up along the DNS hierarchy could in theory end up hogging DNS bandwidth, but practical zones have only very few levels (and Kerberos is not used in reverse DNS) so the number of iterations is very limited. Furthermore, any nuisance would concentrate at the authoritative DNS servers, which are operated by the parties that can insert additional KREALM records to overcome any problems. In short, the burden on DNS should not be aggravated by this iterative approach.

8. Privacy Considerations

It is common to spread service information in DNS, but for internal use this may be considered less desirable. This is why the "service" tag [[Section 6.2](#)], is optional.

Similarly, internal applications may still prefer local definitions for realm names that a client should consider; this specification does not enforce the KREALM record in those situations.

For situations which crossover between realms, the choice between static configuration in files and KDC configuration versus a dynamic configuration in DNS is still a choice; the dynamic option based on DNS publishes more information, but dynamic applications are more likely to desire such information to be publicly and securely available.

The ability to specify administrators in a KREALM record obviously leaks sensitive information about a realm. This should only be done as a deliberate choice, when such information is considered public and useful to the World at large.

9. Security Considerations

This specification defines a mechanism to redirect from a FQDN to a realm name that may not map to that same FQDN, or that define that no realm name exists for the originating FQDN. Publishing such a Kerberos Realm Descriptor is the prerogative of the DNS administrator who is also in charge of publishing the location of the service that is protected by Kerberos. This administrator is generally trusted not to attack the security of a zone; DNSSEC makes this assumption even stronger than unsecured DNS, and this specification does not reduce or expand on that assumption.

When an external attacker would be permitted to spoof a KREALM record in a victim's DNS, then it could be possible for that attacker to

convince the client that the attacker is the authentic provider for the service. Additional spoofing of host name references could then complete the attack. This has been mitigated by requiring DNSSEC for all such KREALM records.

Another angle of attack could be due to suppression of KREALM records, specifically the ones for a host name which have a fallback option at the zone apex. Such attacks could direct a client to rely on information that may form an alternative of lesser security. Such attacks have been mitigated by insisting on signed denials, and by stating that a non-responsive DNS server should not lead to the assumption that one can move up in the DNS hierarchy.

The process of detecting the zone apex relies on the inclusion of a SOA record in each zone apex, and only in the zone apex. Doing this properly is common practice, and it is in the interest of the zone being protected, so no rogue constructs are to be expected for Secure DNS. The presence of a SOA record is not done through an explicit query but rather from inspection of a secure denial on a previously queried domain; this is a secure practice.

The ability to create a KREALM record that references a realm operated under another DNS name introduces a potential of setting flags for that remote realm that may be counter-productive. Given the open-endedness of the IANA registry for tags, problems that this may cause are mitigated by ignoring unknown tags, and treating known tags differently when they are registered as Home Tags; such tags are not processed for references to realms operated under another DNS name.

The specification of administrators in the KREALM record may serve to understand what principal names are a bit more special than others, but in doing so it also defines internal information of a realm; combined with the customary publication of SRV records for a Kadmin service, there is a risk that directed attacks will be mounted at administrator accounts. This might lead to problems when policies are insufficiently tightened, for instance when proxy tickets or constrained delegation are setup to be too permissive. The use of this tag should only be considered under extreme caution.

The specification of administrators in the Home Record may serve to understand who administers principal names for a realm. It should be taken into consideration that the publication path for these "admin" tags may impact their security. Care should also be taken that publishing both principal names and a realm's administration services in DNS suggests focal points for security attacks on a realm. Proper care should be taken to tighten security policies, for instance when proxy tickets or constrained delegation are permitted. The reliance

on the "admin" tag should always be done with the greatest caution, and it may often be required to use it only as a suggestion for posing additional challenges that allow a principal to prove that it actually can add or remove principal names under a realm.

10. IANA Considerations

This specification defines a new "Resource Record (RR) Type", to be registered in IANA registry of Domain Name System (DNS) Parameters". The name of the RRType is KREALM, its value is TBD1 and its meaning is "Kerberos Realm Descriptor".

This specification establishes a new registry with IANA, whose entries are subject to expert review and whose definition must be described in a publicly available specification. The new registry will be known as the "DNS KREALM Tag Registry". Each entry must provide a Yes/No flag to indicate if the tag is a Home Tag, meaning that it may only be interpreted as part of Home Records.

The initial entries for this new registry introduced by this specification are:

+-----+-----+-----+		
Tag name	Home Tag?	Definition
+-----+-----+-----+		
realm	No	[TBD:THIS-SPEC]
service	No	[TBD:THIS-SPEC]
admin	Yes	[TBD:THIS-SPEC]
+-----+-----+-----+		

Tag names are case-sensitive. Registration of new tags is subject to expert review, and a specification must be created as part of its definition.

DISCUSS: Suggestions on the submission process for new tags are requested.

In addition to the foregoing, tag names starting with "x-" are reserved for local and experimental use, for which registration is neither possible nor required. These unregistered tags will not be protected from name clashes.

11. References

11.1. Normative References

- [RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", [RFC 2782](#), DOI 10.17487/RFC2782, February 2000, <<http://www.rfc-editor.org/info/rfc2782>>.
- [RFC3845] Schlyter, J., Ed., "DNS Security (DNSSEC) NextSECure (NSEC) RDATA Format", [RFC 3845](#), DOI 10.17487/RFC3845, August 2004, <<http://www.rfc-editor.org/info/rfc3845>>.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", [RFC 4033](#), DOI 10.17487/RFC4033, March 2005, <<http://www.rfc-editor.org/info/rfc4033>>.
- [RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", [RFC 4120](#), DOI 10.17487/RFC4120, July 2005, <<http://www.rfc-editor.org/info/rfc4120>>.
- [RFC4343] Eastlake 3rd, D., "Domain Name System (DNS) Case Insensitivity Clarification", [RFC 4343](#), DOI 10.17487/RFC4343, January 2006, <<http://www.rfc-editor.org/info/rfc4343>>.
- [RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", [RFC 5155](#), DOI 10.17487/RFC5155, March 2008, <<http://www.rfc-editor.org/info/rfc5155>>.
- [RFC6806] Hartman, S., Ed., Raeburn, K., and L. Zhu, "Kerberos Principal Name Canonicalization and Cross-Realm Referrals", [RFC 6806](#), DOI 10.17487/RFC6806, November 2012, <<http://www.rfc-editor.org/info/rfc6806>>.

11.2. Informative References

- [RFC3655] Wellington, B. and O. Gudmundsson, "Redefinition of DNS Authenticated Data (AD) bit", [RFC 3655](#), DOI 10.17487/RFC3655, November 2003, <<http://www.rfc-editor.org/info/rfc3655>>.
- [RFC4282] Aboba, B., Beadles, M., Arkko, J., and P. Eronen, "The Network Access Identifier", [RFC 4282](#), DOI 10.17487/RFC4282, December 2005, <<http://www.rfc-editor.org/info/rfc4282>>.

- [RFC4592] Lewis, E., "The Role of Wildcards in the Domain Name System", [RFC 4592](#), DOI 10.17487/RFC4592, July 2006, <<http://www.rfc-editor.org/info/rfc4592>>.
- [RFC6840] Weiler, S., Ed. and D. Blacka, Ed., "Clarifications and Implementation Notes for DNS Security (DNSSEC)", [RFC 6840](#), DOI 10.17487/RFC6840, February 2013, <<http://www.rfc-editor.org/info/rfc6840>>.

Author's Address

Rick van Rein
ARPA2.net
Haarlebrink 5
Enschede, Overijssel 7544 WP
The Netherlands

Email: rick@openfortress.nl

