DNS Resolver Module

Overview

The DNS resolver module provides a way for kernel services to make DNS queries by way of requesting a key of key type dns_resolver. These queries are upcalled to userspace through /sbin/request-key.

These routines must be supported by userspace tools dns.upcall, cifs.upcall and request-key. It is under development and does not yet provide the full feature set. The features it does support include:

(*) Implements the dns_resolver key_type to contact userspace.

It does not yet support the following AFS features:

(*) Dns query support for AFSDB resource record.

This code is extracted from the CIFS filesystem.

Compilation

The module should be enabled by turning on the kernel configuration options:

```
CONFIG_DNS_RESOLVER     - tristate "DNS Resolver support"
```

Setting up

To set up this facility, the /etc/request-key.conf file must be altered so that /sbin/request-key can appropriately direct the upcalls. For example, to handle basic dname to IPv4/IPv6 address resolution, the following line should be added:

```
#OP     TYPE            DESC    CO-INFO PROGRAM ARG1 ARG2 ARG3 ...
#====== ============    ======= ======= ============================
create  dns_resolver    *       *       /usr/sbin/cifs.upcall %k
```

To direct a query for query type 'foo', a line of the following should be added before the more general line given above as the first match is the one taken:

```
create  dns_resolver    foo:*   *       /usr/sbin/dns.foo %k
```

Usage

To make use of this facility, one of the following functions that are implemented in the module can be called after doing:

```
   #include <linux/dns_resolver.h>

::

   int dns_query(const char *type, const char *name, size_t namelen,
                 const char *options, char **_result, time_t *_expiry);

This is the basic access function.  It looks for a cached DNS query and if
it doesn't find it, it upcalls to userspace to make a new DNS query, which
may then be cached.  The key description is constructed as a string of the
form::

           [<type>:]<name>

where <type> optionally specifies the particular upcall program to invoke,
and thus the type of query to do, and <name> specifies the string to be
looked up.  The default query type is a straight hostname to IP address
set lookup.

The name parameter is not required to be a NUL-terminated string, and its
length should be given by the namelen argument.

The options parameter may be NULL or it may be a set of options
appropriate to the query type.

The return value is a string appropriate to the query type.  For instance,
for the default query type it is just a list of comma-separated IPv4 and
IPv6 addresses.  The caller must free the result.

The length of the result string is returned on success, and a negative
error code is returned otherwise.  -EKEYREJECTED will be returned if the
```

```
DNS lookup failed.

If _expiry is non-NULL, the expiry time (TTL) of the result will be
returned also.
```

The kernel maintains an internal keyring in which it caches looked up keys. This can be cleared by any process that has the CAP_SYS_ADMIN capability by the use of KEYCTL_KEYRING_CLEAR on the keyring ID.

Reading DNS Keys from Userspace

Keys of dns_resolver type can be read from userspace using keyctl_read() or "keyctl read/print/pipe".

Mechanism

The dnsresolver module registers a key type called "dns_resolver". Keys of this type are used to transport and cache DNS lookup results from userspace.

When dns_query() is invoked, it calls request_key() to search the local keyrings for a cached DNS result. If that fails to find one, it upcalls to userspace to get a new result.

Upcalls to userspace are made through the request_key() upcall vector, and are directed by means of configuration lines in /etc/request-key.conf that tell /sbin/request-key what program to run to instantiate the key.

The upcall handler program is responsible for querying the DNS, processing the result into a form suitable for passing to the keyctl_instantiate_key() routine. This then passes the data to dns_resolver_instantiate() which strips off and processes any options included in the data, and then attaches the remainder of the string to the key as its payload.

The upcall handler program should set the expiry time on the key to that of the lowest TTL of all the records it has extracted a result from. This means that the key will be discarded and recreated when the data it holds has expired.

dns_query() returns a copy of the value attached to the key, or an error if that is indicated instead.

See <file:Documentation/security/keys/request-key.rst> for further information about request-key function.

Debugging

Debugging messages can be turned on dynamically by writing a 1 into the following file:

```
/sys/module/dnsresolver/parameters/debug
```