NAME

local-unbound-anchor - Local-unbound anchor utility.

SYNOPSIS

local-unbound-anchor [**opts**]

DESCRIPTION

Local-unbound-anchor performs setup or update of the root trust anchor for DNSSEC validation. The program fetches the trust anchor with the method from RFC7958 when regular RFC5011 update fails to bring it up to date. It can be run (as root) from the commandline, or run as part of startup scripts. Before you start the *local-unbound*(8) DNS server.

Suggested usage:

```
# in the init scripts.
# provide or update the root anchor (if necessary)
local-unbound-anchor -a "@UNBOUND_ROOTKEY_FILE@"
# Please note usage of this root anchor is at your own risk
# and under the terms of our LICENSE (see source).
#
# start validating resolver
# the unbound.conf contains:
#   auto-trust-anchor-file: "@UNBOUND_ROOTKEY_FILE@"
local-unbound -c unbound.conf
```

This tool provides builtin default contents for the root anchor and root update certificate files.

It tests if the root anchor file works, and if not, and an update is possible, attempts to update the root anchor using the root update certificate. It performs a https fetch of root-anchors.xml and checks the results (RFC7958), if all checks are successful, it updates the root anchor file. Otherwise the root anchor file is unchanged. It performs RFC5011 tracking if the DNSSEC information available via the DNS makes that possible.

It does not perform an update if the certificate is expired, if the network is down or other errors occur.

The available options are:

-a *file*

The root anchor key file, that is read in and written out. Default is @UNBOUND_ROOTKEY_FILE@. If the file does not exist, or is empty, a builtin root key is written to it.

-c *file*
The root update certificate file, that is read in. Default is @UNBOUND_ROOTCERT_FILE@. If the file does not exist, or is empty, a builtin certificate is used.

-l List the builtin root key and builtin root update certificate on stdout.

-u *name*
The server name, it connects to https://name. Specify without https:// prefix. The default is "data.iana.org". It connects to the port specified with -P. You can pass an IPv4 address or IPv6 address (no brackets) if you want.

-S Do not use SNI for the HTTPS connection. Default is to use SNI.

-b *address*
The source address to bind to for domain resolution and contacting the server on https. May be either an IPv4 address or IPv6 address (no brackets).

-x *path*
The pathname to the root-anchors.xml file on the server. (forms URL with -u). The default is /root-anchors/root-anchors.xml.

-s *path*
The pathname to the root-anchors.p7s file on the server. (forms URL with -u). The default is /root-anchors/root-anchors.p7s. This file has to be a PKCS7 signature over the xml file, using the pem file (-c) as trust anchor.

-n *name*
The emailAddress for the Subject of the signer's certificate from the p7s signature file. Only signatures from this name are allowed. default is dnssec@iana.org. If you pass "" then the emailAddress is not checked.

-4 Use IPv4 for domain resolution and contacting the server on https. Default is to use IPv4 and IPv6 where appropriate.

-6 Use IPv6 for domain resolution and contacting the server on https. Default is to use IPv4 and IPv6 where appropriate.

-f *resolv.conf*

Use the given resolv.conf file. Not enabled by default, but you could try to pass /etc/resolv.conf on some systems. It contains the IP addresses of the recursive nameservers to use. However, since this tool could be used to bootstrap that very recursive nameserver, it would not be useful (since that server is not up yet, since we are bootstrapping it). It could be useful in a situation where you know an upstream cache is deployed (and running) and in captive portal situations.

-r *root.hints*

Use the given root.hints file (same syntax as the BIND and Local-unbound root hints file) to bootstrap domain resolution. By default a list of builtin root hints is used. Local-unbound-anchor goes to the network itself for these roots, to resolve the server (-u option) and to check the root DNSKEY records. It does so, because the tool when used for bootstrapping the recursive resolver, cannot use that recursive resolver itself because it is bootstrapping that server.

-R Allow fallback from -f resolv.conf file to direct root servers query. It allows you to prefer local resolvers, but fallback automatically to direct root query if they do not respond or do not support DNSSEC.

-v More verbose. Once prints informational messages, multiple times may enable large debug amounts (such as full certificates or byte-dumps of downloaded files). By default it prints almost nothing. It also prints nothing on errors by default; in that case the original root anchor file is simply left undisturbed, so that a recursive server can start right after it.

-C *unbound.conf*

Debug option to read unbound.conf into the resolver process used.

-P *port*

Set the port number to use for the https connection. The default is 443.

-F Debug option to force update of the root anchor through downloading the xml file and verifying it with the certificate. By default it first tries to update by contacting the DNS, which uses much less bandwidth, is much faster (200 msec not 2 sec), and is nicer to the deployed infrastructure. With this option, it still attempts to do so (and may verbosely tell you), but then ignores the result and goes on to use the xml fallback method.

-h Show the version and commandline option help.

EXIT CODE

This tool exits with value 1 if the root anchor was updated using the certificate or if the builtin root-anchor was used. It exits with code 0 if no update was necessary, if the update was possible with RFC5011 tracking, or if an error occurred.

You can check the exit value in this manner:

```
local-unbound-anchor -a "root.key" || logger "Please check root.key"
```

Or something more suitable for your operational environment.

TRUST

The root keys and update certificate included in this tool are provided for convenience and under the terms of our license (see the LICENSE file in the source distribution or https://github.com/NLnetLabs/unbound/blob/master/LICENSE) and might be stale or not suitable to your purpose.

By running "local-unbound-anchor -l" the keys and certificate that are configured in the code are printed for your convenience.

The build-in configuration can be overridden by providing a root-cert file and a rootkey file.

FILES

@UNBOUND_ROOTKEY_FILE@

The root anchor file, updated with 5011 tracking, and read and written to. The file is created if it does not exist.

@UNBOUND_ROOTCERT_FILE@

The trusted self-signed certificate that is used to verify the downloaded DNSSEC root trust anchor. You can update it by fetching it from https://data.iana.org/root-anchors/icannbundle.pem (and validate it). If the file does not exist or is empty, a builtin version is used.

https://data.iana.org/root-anchors/root-anchors.xml

Source for the root key information.

https://data.iana.org/root-anchors/root-anchors.p7s

Signature on the root key information.

SEE ALSO

unbound.conf(5), *local-unbound*(8).