NAME

nsupdate - Dynamic DNS update utility

SYNOPSIS

nsupdate [**-d**] [**-D**] [[**-g**] | [**-o**] | [**-l**] | [**-y** *[hmac:]keyname:secret*] | [**-k** *keyfile*]] [**-t** *timeout*] [**-u** *udptimeout*] [**-r** *udpretries*] [**-R** *randomdev*] [**-v**] [filename]

DESCRIPTION

nsupdate is used to submit Dynamic DNS Update requests as defined in RFC 2136 to a name server. This allows resource records to be added or removed from a zone without manually editing the zone file. A single update request can contain requests to add or remove more than one resource record.

Zones that are under dynamic control via **nsupdate** or a DHCP server should not be edited by hand. Manual edits could conflict with dynamic updates and cause data to be lost.

The resource records that are dynamically added or removed with **nsupdate** have to be in the same zone. Requests are sent to the zone's master server. This is identified by the MNAME field of the zone's SOA record.

The **-d** option makes **nsupdate** operate in debug mode. This provides tracing information about the update requests that are made and the replies received from the name server.

The **-D** option makes **nsupdate** report additional debugging information to **-d**.

The **-L** option with an integer argument of zero or higher sets the logging debug level. If zero, logging is disabled.

Transaction signatures can be used to authenticate the Dynamic DNS updates. These use the TSIG resource record type described in RFC 2845 or the SIG(0) record described in RFC 2535 and RFC 2931 or GSS-TSIG as described in RFC 3645. TSIG relies on a shared secret that should only be known to **nsupdate** and the name server. Currently, the only supported encryption algorithm for TSIG is HMAC-MD5, which is defined in RFC 2104. Once other algorithms are defined for TSIG, applications will need to ensure they select the appropriate algorithm as well as the key when authenticating each other. For instance, suitable **key** and **server** statements would be added to */etc/named.conf* so that the name server can associate the appropriate secret key and algorithm with the IP address of the client application that will be using TSIG authentication. SIG(0) uses public key cryptography. To use a SIG(0) key, the public key must be stored in a KEY record in a zone served by the name server. **nsupdate** does not read */etc/named.conf*.

GSS-TSIG uses Kerberos credentials. Standard GSS-TSIG mode is switched on with the **-g** flag. A non-standards-compliant variant of GSS-TSIG used by Windows 2000 can be switched on with the **-o** flag.

nsupdate uses the **-y** or **-k** option to provide the shared secret needed to generate a TSIG record for authenticating Dynamic DNS update requests, default type HMAC-MD5. These options are mutually exclusive.

When the **-y** option is used, a signature is generated from [*hmac:*]*keyname:secret*. *keyname*is the name of the key, and *secret* is the base64 encoded shared secret. Use of the **-y** option is discouraged because the shared secret is supplied as a command line argument in clear text. This may be visible in the output from **ps(1)** or in a history file maintained by the user's shell.

With the **-k** option, **nsupdate** reads the shared secret from the file *keyfile*. Keyfiles may be in two formats: a single file containing a *named.conf*-format **key** statement, which may be generated automatically by **ddns-confgen**, or a pair of files whose names are of the format *K{name}.+157.+{random}.key* and *K{name}.+157.+{random}.private*, which can be generated by **dnssec-keygen**. The **-k** may also be used to specify a SIG(0) key used to authenticate Dynamic DNS update requests. In this case, the key specified is not an HMAC-MD5 key.

nsupdate can be run in a local-host only mode using the **-l** flag. This sets the server address to

localhost (disabling the **server** so that the server address cannot be overridden). Connections to the local server will use a TSIG key found in */var/run/named/session.key*, which is automatically generated by **named** if any local master zone has set **update-policy** to **local**. The location of this key file can be overridden with the **-k** option.

By default, **nsupdate** uses UDP to send update requests to the name server unless they are too large to fit in a UDP request in which case TCP will be used. The **-v** option makes **nsupdate** use a TCP connection. This may be preferable when a batch of update requests is made.

The **-p** sets the default port number to use for connections to a name server. The default is 53.

The **-t** option sets the maximum time an update request can take before it is aborted. The default is 300 seconds. Zero can be used to disable the timeout.

The **-u** option sets the UDP retry interval. The default is 3 seconds. If zero, the interval will be computed from the timeout interval and number of UDP retries.

The **-r** option sets the number of UDP retries. The default is 3. If zero, only one update request will be made.

The **-R** *randomdev* option specifies a source of randomness. If the operating system does not provide a */dev/random* or equivalent device, the default source of randomness is keyboard input. *randomdev* specifies the name of a character device or file containing random data to be used instead of the default. The special value *keyboard* indicates that keyboard input should be used. This option may be specified multiple times.

INPUT FORMAT

nsupdate reads input from *filename* or standard input. Each command is supplied on exactly one line of input. Some commands are for administrative purposes. The others are either update instructions or prerequisite checks on the contents of the zone. These checks set conditions that some name or set of resource records (RRset) either exists or is absent from the zone. These conditions must be met if the entire update request is to succeed. Updates will be rejected if the tests for the prerequisite conditions fail.

Every update request consists of zero or more prerequisites and zero or more updates. This allows a suitably authenticated update request to proceed if some specified resource records are present or missing from the zone. A blank input line (or the **send** command) causes the accumulated commands to be sent as one Dynamic DNS update request to the name server.

The command formats and their meaning are as follows:

server {servername} [port]
: Sends all dynamic update requests to the name server *servername*. When no server statement is provided, **nsupdate** will send updates to the master server of the correct zone. The MNAME field of that zone's SOA record will identify the master server for that zone. *port* is the port number on *servername* where the dynamic update requests get sent. If no port number is specified, the default DNS port number of 53 is used.

local {address} [port]
: Sends all dynamic update requests using the local *address*. When no local statement is provided, **nsupdate** will send updates using an address and port chosen by the system. *port* can additionally be used to make requests come from a specific port. If no port number is specified, the system will assign one.

zone {zonename}
: Specifies that all updates are to be made to the zone *zonename*. If no *zone* statement is provided, **nsupdate** will attempt determine the correct zone to update based on the rest of the input.

class {classname}
: Specify the default class. If no *class* is specified, the default class is *IN*.

ttl {seconds}

Specify the default time to live for records to be added. The value *none* will clear the default ttl.

key {name} {secret}
Specifies that all updates are to be TSIG-signed using the *keyname keysecret* pair. The **key** command overrides any key specified on the command line via **-y** or **-k**.

gsstsig
Use GSS-TSIG to sign the updated. This is equivalent to specifying **-g** on the commandline.

oldgsstsig
Use the Windows 2000 version of GSS-TSIG to sign the updated. This is equivalent to specifying **-o** on the commandline.

realm {[realm_name]}
When using GSS-TSIG use *realm_name* rather than the default realm in *krb5.conf*. If no realm is specified the saved realm is cleared.

[**prereq**] **nxdomain**{domain-name}
Requires that no resource record of any type exists with name *domain-name*.

[**prereq**] **yxdomain**{domain-name}
Requires that *domain-name* exists (has as at least one resource record, of any type).

[**prereq**] **nxrrset**{domain-name} [class] {t ype}
Requires that no resource record exists of the specified *type*, *class* and *domain-name*. If *class* is omitted, IN (internet) is assumed.

[**prereq**] **yxrrset**{domain-name} [class] {t ype}
This requires that a resource record of the specified *type*, *class* and *domain-name* must exist. If *class* is omitted, IN (internet) is assumed.

[**prereq**] **yxrrset**{domain-name} [class] {t ype} {data...}
The *data* from each set of prerequisites of this form sharing a common *type*, *class*, and *domain-name* are combined to form a set of RRs. This set of RRs must exactly match the set of RRs existing in the zone at the given *type*, *class*, and *domain-name*. The *data* are written in the standard text representation of the resource record's RDATA.

[**update**] **del[ete]**{domain-name} [ttl] [class] [t ype [data...]]
Deletes any resource records named *domain-name*. If *type* and *data* is provided, only matching resource records will be removed. The internet class is assumed if *class* is not supplied. The *ttl* is ignored, and is only allowed for compatibility.

[**update**] **add**{domain-name} {ttl} [class] {t ype} {data...}
Adds a new resource record with the specified *ttl*, *class* and *data*.

show
Displays the current message, containing all of the prerequisites and updates specified since the last send.

send
Sends the current message. This is equivalent to entering a blank line.

answer
Displays the answer.

debug
Turn on debugging.

Lines beginning with a semicolon are comments and are ignored.

EXAMPLES

The examples below show how **nsupdate** could be used to insert and delete resource records from the **example.com** zone. Notice that the input in each example contains a trailing blank line so that a group of commands are sent as one dynamic update request to the master name server for

example.com.

```
# nsupdate
> update delete oldhost.example.com A
> update add newhost.example.com 86400 A 172.16.1.1
> send
```

Any A records for **oldhost.example.com** are deleted. And an A record for **newhost.example.com** with IP address 172.16.1.1 is added. The newly-added record has a 1 day TTL (86400 seconds).

```
# nsupdate
> prereq nxdomain nickname.example.com
> update add nickname.example.com 86400 CNAME somehost.example.com
> send
```

The prerequisite condition gets the name server to check that there are no resource records of any type for **nickname.example.com**. If there are, the update request fails. If this name does not exist, a CNAME for it is added. This ensures that when the CNAME is added, it cannot conflict with the long-standing rule in RFC 1034 that a name must not exist as any other record type if it exists as a CNAME. (The rule has been updated for DNSSEC in RFC 2535 to allow CNAMEs to have RRSIG, DNSKEY and NSEC records.)

FILES

/etc/resolv.conf
used to identify default name server

/var/run/named/session.key
sets the default TSIG key for use in local-only mode

K{name}.+157.+{random}.key
base-64 encoding of HMAC-MD5 key created by **dnssec-keygen(8)**.

K{name}.+157.+{random}.private
base-64 encoding of HMAC-MD5 key created by **dnssec-keygen(8)**.

SEE ALSO

RFC 2136, RFC 3007, RFC 2104, RFC 2845, RFC 1034, RFC 2535, RFC 2931, **named(8)**, **ddns-confgen(8)**, **dnssec-keygen(8)**.

BUGS

The TSIG key is redundantly stored in two separate files. This is a consequence of nsupdate using the DST library for its cryptographic operations, and may change in future releases.

COPYRIGHT