NAME

dnssec-cds − change DS records for a child zone based on CDS/CDNSKEY

SYNOPSIS

dnssec−cds [**−a** alg...] [**−c** class] [**−D**] {**−d** dsset−file} {**−f** child−file} [**−i**[extension]] [−s** start−time] [**−T** ttl] [**−u**] [**−v** level] [**−V**] {domain}

DESCRIPTION

The **dnssec−cds** command changes DS records at a delegation point based on CDS or CDNSKEY records published in the child zone. If both CDS and CDNSKEY records are present in the child zone, the CDS is preferred. This enables a child zone to inform its parent of upcoming changes to its key−signing keys (KSKs); by polling periodically with **dnssec−cds**, the parent can keep the DS records up−to−date and enable automatic rolling of KSKs.

Two input files are required. The **−f child−file** option specifies a file containing the child's CDS and/or CDNSKEY records, plus RRSIG and DNSKEY records so that they can be authenticated. The **−d path** option specifies the location of a file containing the current DS records. For example, this could be a **dsset−** file generated by **dnssec−signzone** <#std-iscman-dnssec-signzone>, or the output of **dnssec−dsfromkey** <# std-iscman-dnssec-dsfromkey>, or the output of a previous run of **dnssec−cds**.

The **dnssec−cds** command uses special DNSSEC validation logic specified by **RFC 7344** <https:// datatracker.ietf.org/doc/html/rfc7344.html>. It requires that the CDS and/or CDNSKEY records be validly signed by a key represented in the existing DS records. This is typically the pre−existing KSK.

For protection against replay attacks, the signatures on the child records must not be older than they were on a previous run of **dnssec−cds**. Their age is obtained from the modification time of the **dsset−** file, or from the **−s** option.

To protect against breaking the delegation, **dnssec−cds** ensures that the DNSKEY RRset can be verified by every key algorithm in the new DS RRset, and that the same set of keys are covered by every DS digest type.

By default, replacement DS records are written to the standard output; with the **−i** option the input file is overwritten in place. The replacement DS records are the same as the existing records, when no change is required. The output can be empty if the CDS/CDNSKEY records specify that the child zone wants to be insecure.

Warning:
Be careful not to delete the DS records when **dnssec−cds** fails!

Alternatively, :option`dnssec−cds −u` writes an **nsupdate** <#std-iscman-nsupdate> script to the standard output. The **−u** and **−i** options can be used together to maintain a **dsset−** file as well as emit an **nsupdate** <# std-iscman-nsupdate> script.

OPTIONS

−a algorithm

When converting CDS records to DS records, this option specifies the acceptable digest algorithms. This option can be repeated, so that multiple digest types are allowed. If none of the CDS records use an acceptable digest type, **dnssec−cds** will try to use CDNSKEY records instead; if there are no CDNSKEY records, it reports an error.

When converting CDNSKEY records to DS records, this option specifies the digest algorithm to use. It can be repeated, so that multiple DS records are created for each CDNSKEY records.

The algorithm must be one of SHA−1, SHA−256, or SHA−384. These values are case−insensitive, and the hyphen may be omitted. If no algorithm is specified, the default is SHA−256 only.

−c class
This option specifies the DNS class of the zones.

−D This option generates DS records from CDNSKEY records if both CDS and CDNSKEY records are present in the child zone. By default CDS records are preferred.

−d path
This specifies the location of the parent DS records. The path can be the name of a file containing the DS records; if it is a directory, **dnssec−cds** looks for a **dsset−** file for the domain inside the directory.

To protect against replay attacks, child records are rejected if they were signed earlier than the modification time of the **dsset−** file. This can be adjusted with the **−s** option.

−f child−file
This option specifies the file containing the child's CDS and/or CDNSKEY records, plus its DNSKEY records and the covering RRSIG records, so that they can be authenticated.

The examples below describe how to generate this file.

−i extension
This option updates the **dsset−** file in place, instead of writing DS records to the standard output.

There must be no space between the **−i** and the extension. If no extension is provided, the old **dsset−** is discarded. If an extension is present, a backup of the old **dsset−** file is kept with the extension appended to its filename.

To protect against replay attacks, the modification time of the **dsset−** file is set to match the signature inception time of the child records, provided that it is later than the file's current modification time.

−s start−time
This option specifies the date and time after which RRSIG records become acceptable. This can be either an absolute or a relative time. An absolute start time is indicated by a number in YYYYMMDDHHMMSS notation; 20170827133700 denotes 13:37:00 UTC on August 27th, 2017. A time relative to the **dsset−** file is indicated with **−N**, which is N seconds before the file modification time. A time relative to the current time is indicated with **now+N**.

If no start−time is specified, the modification time of the **dsset−** file is used.

−T ttl This option specifies a TTL to be used for new DS records. If not specified, the default is the TTL of the old DS records. If they had no explicit TTL, the new DS records also have no explicit TTL.

−u This option writes an **nsupdate** <#std-iscman-nsupdate> script to the standard output, instead of printing the new DS reords. The output is empty if no change is needed.

Note: The TTL of new records needs to be specified: it can be done in the original **dsset−** file, with the **−T** option, or using the **nsupdate** <#std-iscman-nsupdate> **ttl** command.

−V This option prints version information.

−v level
This option sets the debugging level. Level 1 is intended to be usefully verbose for general users; higher levels are intended for developers.

domain
This indicates the name of the delegation point/child zone apex.

EXIT STATUS

The **dnssec−cds** command exits 0 on success, or non−zero if an error occurred.

If successful, the DS records may or may not need to be changed.

EXAMPLES

Before running **dnssec−signzone** <#std-iscman-dnssec-signzone>, ensure that the delegations are up−to−date by running **dnssec−cds** on every **dsset−** file.

To fetch the child records required by **dnssec−cds**, invoke **dig** <#std-iscman-dig> as in the script below. It is acceptable if the **dig** <#std-iscman-dig> fails, since **dnssec−cds** performs all the necessary checking.

```
for f in dsset−*
do
    d=${f#dsset−}
    dig +dnssec +noall +answer $d DNSKEY $d CDNSKEY $d CDS |
    dnssec−cds −i −f /dev/stdin −d $f $d
done
```

When the parent zone is automatically signed by **named** <#std-iscman-named>, **dnssec−cds** can be used with **nsupdate** <#std-iscman-nsupdate> to maintain a delegation as follows. The**dsset−** file allo ws the script to avoid having to fetch and validate the parent DS records, and it maintains the replay attack protection time.

```
dig +dnssec +noall +answer $d DNSKEY $d CDNSKEY $d CDS |
dnssec−cds −u −i −f /dev/stdin −d $f $d |
nsupdate −l
```

SEE ALSO

dig(1) <#std-iscman-dig>, **dnssec−settime(8)** <#std-iscman-dnssec-settime>, **dnssec−signzone(8)** <#std-iscman-dnssec-signzone>, **nsupdate(1)** <#std-iscman-nsupdate>, BIND 9 Administrator Reference Manual, **RFC 7344** <https://datatracker.ietf.org/doc/html/rfc7344.html>.

Author

Internet Systems Consortium

Copyright