NAME

ksi verify - Guardtime command-line tool for verifying KSI signatures.

SYNOPSIS

ksi verify -i *in.ksig* [**-f** *data*] [*more options*]

ksi verify --ver-int -i *in.ksig* [**-f** *data*] [*more options*]

ksi verify --ver-cal -i *in.ksig* [**-f** *data*] **-X** *URL* [**--ext-user** *user* **--ext-key** *key*] [*more options*]

ksi verify --ver-key -i *in.ksig* [**-f** *data*] **-P** *URL* [**--cnstr** *oid=value*]... [**-V** *cert.pem*]... [**-W** *dir*]... [*more options*]

ksi verify --ver-pub -i *in.ksig* [**-f** *data*] **--pub-str** *pubstr* [**-x -X** *URL* [**--ext-user** *user* **--ext-key** *key*]] [*more options*]

ksi verify --ver-pub -i *in.ksig* [**-f** *data*] **-P** *URL* [**--cnstr** *oid=value*]... [**-V** *cert.pem*]... [**-W** *dir*]... [**-x -X** *URL* [**--ext-user** *user* **--ext-key** *key*]] [*more options*]

DESCRIPTION

This is a KSI signature verification tool. There are four main verification policies related to a specific command line option that can be applied to a KSI signature. Signature can be verified with or without the file the signature was created for or with its pre calculated hash value imprint. Possible verification polices are:

- Internal verification. Only signatures internal consistency is checked and no verification against any trust anchor is performed. This step is always performed during other polices. No external resources are needed.
- Calendar-based verification. Signature is verified against calendar database. Verification is done by checking if the calculated output hash value of the aggregation hash chain exists in the calendar database at the same time the signature was created. Access to a trusted KSI extender is needed.
- Key-based verification. Signature must contain a calendar hash chain and a calendar authentication record that can be verified against the signing certificates. To be able to perform key-based verification user must have an access to a trusted KSI publications file with signing certificates in it.
- Publication-based verification. Signature must be extended (see **ksi-extend**(1)) to a time of publication and contain a publication record. Verification is done by checking the signatures internal consistency and comparing the signatures publication record with publication data retrieved from publications file or publication string given from printed media. Trusted publications file or a copy of a printed media with corresponding published publication string in it is needed.

It must be noted that only publication-based verification should be used in long term as it does not rely on any keys and trusted services. The other polices can be used temporarily when the signature is created and there is not yet a publication to extend the signature to.

OPTIONS

--ver-int
: Perform internal verification.

--ver-cal
: Perform calendar-based verification (use extending service).

--ver-key
: Perform key-based verification.

--ver-pub
: Perform publication-based verification.

-i *sig* KSI signature file to be verified. Use '-' as file name to read signatures file from *stdin*.

-f *data* Files path to file to be hashed or data hash imprint to extract the hash value that is going to be verified. Hash format: <alg>:<hash in hex>. Use '-' as file name to read data to be hashed from *stdin*. Call **ksi -h** to get the list of supported hash algorithms.

-X *URL*
Specify extending service URL.

--ext-user *str*
User name for extending service.

--ext-key *str*
HMAC Key for extending service.

-x Permit to use extender when using publication based verification.

--pub-str *str*
Specify a publication string to verify with.

-P *URL* Specify publications file URL (or file with URI scheme 'file://').

--cnstr *oid=value*
OID and its expected value to verify publications file PKI signature. At least one constraint must be defined to be able to verify publications file but it is possible to define more. All values from lower priority source are ignored, where default configurations file is the lowest and command-line is the highest.

-V *file* Specify an OpenSSL-style trust store file for publications file verification. All values from lower priority source are ignored, where default configurations file is the lowest and command-line is the highest.

-W *dir* Specify an OpenSSL-style trust store directory for publications file verification. All values from lower priority source are ignored, where default configurations file is the lowest and command-line is the highest.

-d Print detailed information about processes and errors to *stderr*.

--dump
Dump signature and document hash being verified to *stdout*.

--conf *file*
Specify a configurations file to override default service information. It must be noted that service info from command-line will override the configurations file. See **ksi**(1) and **ksi-conf**(5) for more information.

--log *file*
Write libksi log into file. Use '-' as file name to redirect log to *stdout*.

EXIT STATUS

See **ksi**(1) for more information.

EXAMPLES

In the following examples it is assumed that default service URLs are defined (see **ksi**(1) for more information). Signature files with extension **.ksig** are not and files with extension **.ext.ksig** are extended signatures.

1 To perform internal verification on KSI signature file *test.ksig* and document *test* call:
ksi verify --ver-int -i *test.ksig* **-f** *test*

2 To perform key based verification on freshly created KSI signature file *test.ksig* and document hash call:
ksi verify --ver-key -i *test.ksig* **-f** *SHA-256:c8ef6d57ac28d1b4e95a513959f5fcdd0688380a43d601a5ace1d2e96884690a*

3 To perform calendar based verification on KSI signature file *test.ksig* call:
ksi verify --ver-cal -i *test.ksig*

4 To perform publication based verification (printed in public media) on KSI signature file *test.ext.ksig* call:
ksi verify --ver-pub -i *test.ext.ksig* **--pub-str** *AAAAAA-CWYEKQ-AAIYPA-UJ4GRT-HXMFBE-OTB4AB-XH3PT3-KNIKGV-PYCJXU-HL2TN4-RG6SCC-3ZGSBM*

5 To perform publication based verification (publication from publications file) on KSI signature file *test.ext.ksig* call:

ksi verify --ver-pub -i *test.ext.ksig*

6 To perform publication based verification on not extended KSI signature file *test.ksig* call:

ksi verify --ver-pub -i *test.ksig* **-x**

7 To perform verification on KSI signature *test.ksig* as possible and dump its content call:

ksi verify -i *test.ksig* **--dump**

ENVIRONMENT

Environment variable **KSI_CONF** can be defined to set default KSI configurations file. See **ksi**(1) and **ksi-conf**(5) for more information.

AUTHOR

Guardtime AS, http://www.guardtime.com/

SEE ALSO

ksi(1), **ksi-sign**(1), **ksi-extend**(1), **ksi-pubfile**(1), **ksi-conf**(5)