NAME

ksitool - Guardtime command-line tool to access KSI service.

SYNOPSIS

Sign data file, stream (stdin) or hash:
ksitool -s -f *file* | [**-F** *hash*] **-o** *out.ksig* [*more options*]

Extend signature:
ksitool -x -i *sig.ksig* **-o** *ext.ksig* [**-T**] [*more options*]

Download and verify publication file:
ksitool -p -o *pubfile* **--cnstr** *oid=value ...* [*more options*]

Dump publication file:
ksitool -p -d -P *url* | [**-b** *pubfile*]

Download publication file:
ksitool -p -o *pubfile* **--cnstr** *oid=value ...* [*more options*]

Create publication string:
ksitool -p -T [*more options]*

Verify signature:
ksitool -v -i *sig.ksig* [**-f** *file* | **-F** *hash*] [**-x** | **-b** *pubfile*] [*more options*]

Verify signature with publications string:
ksitool -v -i *sig.ksig* **--ref** *pubstr* [**-f** *file* | **-F** *hash*] [**-x**] [*more options*]

Verify publication file:
ksitool -v -b *pubfile* **--cnstr** *oid=value ...* [*more options*]

DESCRIPTION

This is a general signing and signature verification tool for Guardtime Keyless Signature services.

OPTIONS

Tasks:

-p Use for downloading publication file and creating publication string.

-s, --sign
Use for signing document, stream (stdin) or hash.

-v, --verify
Use for signature and publication file verification.

-x, --extend
Use for online verification or signature extending (permit extending when verifying signature with publication string).

Input/output:

-b *file* Use specified publication file.

-f *file* File to be signed or verified. *file* can be '**-**' to sign a stream (stdin).

-F *hash*
Data hash to be signed or verified. Hash format: <alg>:<hash in hex>.

-H *alg* Hash algorithm used to hash the file to be signed. Use when signing file (**-s -f**).

-i *file* Input signature token file to be extended or verified.

-o *file* Output file name to store signature token or publication file.

-D, --dataout *file*
Output file name to store data from stream (stdin).

--ref *str*
Publication string as base 32 encoded string. Use with **-v**, add **-x** to permit extending.

-T *int*
Specify a publication time to extend to (use with **-x**) or a time to create a new publication code for (use with **-p**) as the number of seconds since 1970-01-01 00:00:00 UTC or time string formatted as "YYYY-MM-DD hh:mm:ss".

Details:

-d Print detailed information.

--log *file*
Dump KSI log into file.

-n Print signer Name (identity).

--nowarn
Silence warning messages.

-r Print publication References (use with -vx).

--silent Silence info and warning messages.

-t Print service Timing in ms.

Configuration:

-c *int* Network transfer timeout in seconds, after successful Connect.

-C *int* Network Connect timeout in seconds (is not supported with tcp client).

--cnstr *oid=value*
use 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. If value part contains spaces use " " to wrap its contents. For common OID's there are string representations: '**email**' for 1.2.840.113549.1.9.1, '**country**' for 2.5.4.6, '**org**' for 2.5.4.10 and '**common_name**' for 2.5.4.3. Example **--cnstr** *2.5.4.6=EE* **--cnstr** *org="Guardtime AS"*.

--inc *file*
Use configuration file containing command-line parameters. Parameter must be written line by line.

--pass *str*
Password for authentication.

-P *url* Specify publication file URL.

-S *url* Specify Signing service URL.

--user *str*
User name for authentication.

-V *file* Use specified OpenSSL-style trust store file for publication file verification. Can have multiple values (-V <file 1> -V <file 2>).

-W *dir* Use specified OpenSSL-style trust store directory for publication file verification.

-X *url* Specify verification (eXtending) service URL.

Help:

-h, --help
print ksitool help.

EXIT STATUS

0 Exit success. Returned if everything is OK.

1 Exit failure. A general failure occurred.

3 Invalid command-line parameter. The content or format of command-line parameter is invalid. Also a parameter may be missing.

4 Invalid format. Input data to KSI library is invalid, for example signature or publication file format is invalid.

5 Network error. Ksitool is unable to connect to the service, connection is timed out or an HTTP error was returned from the service url.

6 Verification error. Verification of signature or document hash failed.

7 Extending error. Error in extending a signature or an error was returned by extender.

8 Aggregation error. Error returned by aggregator.

9 Input / output error. Unable to write or read file.

10 Cryptographic error. Error may be generated due to untrusted or unavailable hash algorithm or by an invalid PKI signature or untrusted certificate.

11 HMAC error. HMAC of aggregation or extend response does not match.

12 No privileges. Operating system did not grant privileges to perform an operation.

13 System out of memory.

14 Authentication error. Aggregation or extending service did not accept user identification parameters.

EXAMPLES

In the following examples it is assumed that default service urls are defined as environment variables. Read example 1 to learn how to define service urls.

1 To use ksitool, service urls must be specified. It can be done via environment variables, command-line parameters or a configuration file.

1.1 To define default urls as environment variables, KSI_AGGREGATOR, KSI_EXTENDER and KSI_PUBFILE must be defined as shown below:

KSI_AGGREGATOR=*url=http://test.com:3333/gt-signingservice pass=test_pass user=test_user*
KSI_EXTENDER=*url=http://test.com:8010/gt-extendingservice pass=test_pass user=test_user*
KSI_PUBFILE=*url=http://verify.guardtime.com/ksi-publications.bin 1.2.840.113549.1.9.1=publications@guardtime.com 2.5.4.10="Symantec Corporation"*

1.2 To define service urls on command-line or via configuration file, following parameters must be defined:
-X *http://test.com:8010/gt-extendingservice*
-S *http://test.com:3333/gt-signingservice*
--user *test_user*
--pass *test_pass*

1.3 To use a configuration file, parameters must be written on separate lines, into a file, as in the example above. Configuration file *conf* must be included using option:
--inc *conf*

2 To sign a file *file* and save signature to *sig.ksig* call:
ksitool -s -f *file* **-o** *sig.ksig*

2.1 To sign a stream (stdin), save data from stream to *file* and save signature to *sig.ksig* call:
ksitool -s -f - -D *file* **-o** *sig.ksig*

2.2 To sign a stream (stdin) and save signature to *sig.ksig* without saving data from stream call:
ksitool -s -f - -o *sig.ksig*

3 To sign a data hash (hashed with SHA256) and save the resulting signature to file *sig.ksig* call:
ksitool -s -o *sig.ksig* **-F** *SHA-256:c8ef6d57ac28d1b4e95a513959f5fcdd0688380a43d601a5ace1d2e96884690a*

4 To sign a data file *file* with non-default algorithm *SHA1* call:
ksitool -s -f *file* **-H** *SHA1* **-o** *sig.ksig*

5 To verify a signature *sig.ksig* and file *file* it belongs to call:
ksitool -v -i *sig.ksig* **-f** *file*

6 To verify a signature *sig.ksig* and hash it belongs to call:
ksitool -v -i *sig.ksig* **-F** *SHA-256:c8ef6d57ac28d1b4e95a513959f5fcdd0688380a43d601a5ace1d2e96884690a*

7 To verify a signature *sig.ksig* using online verification service call:
ksitool -vx -i *sig.ksig*

8 To extend a signature *sig.ksig* and save it as *ext.ksig* call:
ksitool -x -i *sig.ksig* **-o** *ext.ksig*

9 To verify an extended signature *ext.ksig* against publication from printed media call:
ksitool -v -i *ext.ksig* **--ref** *AAAAAA-CT5VGY-AAPUCF-L3EKCC-NRSX56-AXIDFL-VZJQK4-WDCPOE-3KIWGB-XGPPM3-O5BIMW-REOVR4*

10 To download a publication file *pubfile* call:
ksitool -p -o *pubfile*

11 To verify publication file *pubfile* call:
ksitool -v -b *pubfile*

12 To create a publication string call:
ksitool -p -T *"2015-10-15 00:00:00"*

ENVIRONMENT

Default service access URL-s:

To define default URL-s, they must be defined as environment variables. For aggregator and extender service, define environment variables **KSI_AGGREGATOR** and **KSI_EXTENDER** with content *'url=<url> pass=<pass> user=<user>'*. Only url part is mandatory: user and pass can be left undefined if anonymous access is allowed by the service. Default *<pass>* and *<user>* is *'anon'*.

For publications file, define **KSI_PUBFILE** with content *'url=<url> <constraint> <constraint> ...'*. Constraint is formatted as *<OID>="<value>"* where "" can be omitted if 'value' does not contain any white-space characters. Publications file url is mandatory but constraints are not if at least one constraint is defined on command-line (see **--cnstr**).

Using includes (**--inc**) or defining urls on command-line will override defaults.

AUTHOR

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