NAME

ovs−pki − OpenFlow public key infrastructure management utility

SYNOPSIS

Each command takes the form:

ovs−pki [*options*] *command* [*args*]

The implemented commands and their arguments are:
ovs−pki init
ovs−pki req *name*
ovs−pki sign *name* [*type*]
ovs−pki req+sign *name* [*type*]
ovs−pki verify *name* [*type*]
ovs−pki fingerprint *file*
ovs−pki self−sign *name*

Each *type* above is a certificate type, either **switch** (default) or **controller**.

The available options are:
[**−k** *type* | **−−key=***type*]
[**−B** *nbits* | **−−bits=***nbits*]
[**−D** *file* | **−−dsaparam=***file*]
[**−b** | **−−batch**]
[**−f** | **−−force**]
[**−d** *dir* | **−−dir=***dir*]
[**−l** *file* | **−−log=***file*]
[**−h** | **−−help**]

Some options do not apply to every command.

DESCRIPTION

The **ovs−pki** program sets up and manages a public key infrastructure for use with OpenFlow. It is intended to be a simple interface for organizations that do not have an established public key infrastructure. Other PKI tools can substitute for or supplement the use of **ovs−pki**.

ovs−pki uses **openssl**(1) for certificate management and key generation.

OFFLINE COMMANDS

The following **ovs−pki** commands support manual PKI administration:

init Initializes a new PKI (by default in directory **/usr/local/share/openvswitch/pki**) and populates it with a pair of certificate authorities for controllers and switches.

This command should ideally be run on a high−security machine separate from any OpenFlow controller or switch, called the CA machine. The files **pki/controllerca/cacert.pem** and **pki/switchca/cacert.pem** that it produces will need to be copied over to the OpenFlow switches and controllers, respectively. Their contents may safely be made public.

By default, **ovs−pki** generates 2048−bit RSA keys. The **−B** or **−−bits** option (see below) may be used to override the key length. The **−k dsa** or **−−key=dsa** option may be used to use DSA in place of RSA. If DSA is selected, the **dsaparam.pem** file generated in the new PKI hierarchy must be copied to any machine on which the **req** command (see below) will be executed. Its

contents may safely be made public.

Other files generated by **init** may remain on the CA machine. The files **pki/controllerca/private/cakey.pem** and **pki/switchca/private/cakey.pem** have particularly sensitive contents that should not be exposed.

req *name*

Generates a new private key named *name*−**privkey.pem** and corresponding certificate request named *name*−**req.pem**. The private key can be intended for use by a switch or a controller.

This command should ideally be run on the switch or controller that will use the private key to identify itself. The file *name*−**req.pem** must be copied to the CA machine for signing with the **sign** command (below).

This command will output a fingerprint to stdout as its final step. Write down the fingerprint and take it to the CA machine before continuing with the **sign** step.

When RSA keys are in use (as is the default), **req**, unlike the rest of **ovs−pki**'s commands, does not need access to a PKI hierarchy created by **ovs−pki init**. The **−B** or **−−bits** option (see below) may be used to specify the number of bits in the generated RSA key.

When DSA keys are used (as specified with **−−key=dsa**), **req** needs access to the **dsaparam.pem** file created as part of the PKI hierarchy (but not to other files in that tree). By default, **ovs−pki** looks for this file in **/usr/local/share/openvswitch/pki/dsaparam.pem**, but the **−D** or **−−dsaparam** option (see below) may be used to specify an alternate location.

name−**privkey.pem** has sensitive contents that should not be exposed. *name*−**req.pem** may be safely made public.

sign *name* [*type*]

Signs the certificate request named *name*−**req.pem** that was produced in the previous step, producing a certificate named *name*−**cert.pem**. *type*, either **switch** (default) or **controller**, indicates the use for which the key is being certified.

This command must be run on the CA machine.

The command will output a fingerprint to stdout and request that you verify that it is the same fingerprint output by the **req** command. This ensures that the request being signed is the same one produced by **req**. (The **−b** or **−−batch** option suppresses the verification step.)

The file *name*−**cert.pem** will need to be copied back to the switch or controller for which it is intended. Its contents may safely be made public.

req+sign *name* [*type*]

Combines the **req** and **sign** commands into a single step, outputting all the files produced by each. The *name*−**privkey.pem** and *name*−**cert.pem** files must be copied securely to the switch or controller. *name*−**privkey.pem** has sensitive contents and must not be exposed in transit. Afterward, it should be deleted from the CA machine.

This combined method is, theoretically, less secure than the individual steps performed separately on two different machines, because there is additional potential for exposure of the private key. However, it is also more convenient.

verify *name* [*type*]
Verifies that *name*−**cert.pem** is a valid certificate for the given *type* of use, either **switch** (default) or **controller**. If the certificate is valid for this use, it prints the message ‘‘*name*−**cert.pem**: OK’’; otherwise, it prints an error message.

fingerprint *file*
Prints the fingerprint for *file*. If *file* is a certificate, then this is the SHA−1 digest of the DER encoded version of the certificate; otherwise, it is the SHA−1 digest of the entire file.

self−sign *name*
Signs the certificate request named *name*−**req.pem** using the private key *name*−**privkey.pem**, producing a self-signed certificate named *name*−**cert.pem**. The input files should have been produced with **ovs−pki req**.

Some controllers accept such self-signed certificates.

OPTIONS

−k *type*
−−key=*type*
For the **init** command, sets the public key algorithm to use for the new PKI hierarchy. For the **req** and **req+sign** commands, sets the public key algorithm to use for the key to be generated, which must match the value specified on **init**. With other commands, the value has no effect.

The *type* may be **rsa** (the default) or **dsa**.

−B *nbits*
−−bits=*nbits*
Sets the number of bits in the key to be generated. When RSA keys are in use, this option affects only the **init**, **req**, and **req+sign** commands, and the same value should be given each time. With DSA keys are in use, this option affects only the **init** command.

The value must be at least 1024. The default is 2048.

−D *file*
−−dsaparam=*file*
Specifies an alternate location for the **dsaparam.pem** file required by the **req** and **req+sign** commands. This option affects only these commands, and only when DSA keys are used.

The default is **dsaparam.pem** under the PKI hierarchy.

−b
−−batch
Suppresses the interactive verification of fingerprints that the **sign** command by default requires.

−d *dir*
−−dir=*dir*
Specifies the location of the PKI hierarchy to be used or created by the command (default: **/usr/local/share/openvswitch/pki**). All commands, except **req**, need access to a PKI hierarchy.

−f

−−force
By default, **ovs−pki** will not overwrite existing files or directories. This option overrides this behavior.

−l *file*
−−log=*file*
Sets the log file to *file*. Default: **/usr/local/var/log/openvswitch/ovs−pki.log**.

−h
−−help Prints a help usage message and exits.

SEE ALSO

ovs−controller(8).