secret: Share Sensitive Information in R Packages

2017-06-17

Usage

Load the package:

```
library(secret)
```

Set up your keys:

Ensure you know the location of your public and private keys. In Linux this is usually the folder `~/.ssh` , so on Windows you may want to choose the same folder.

By default, the package looks for your private key at

1. `~/.ssh/id_rsa`
2. `~/.ssh/id_rsa.pem` .

You can change this default by setting an environment variable `USER_KEY` :

```
# This is optional - only do this if you want to change the default location
Sys.setenv(USER_KEY = "path/to/private/key")
```

Test that the package can read your key. This might fail if you don't have a key at `~/.ssh/id_rsa` , or if your private key has a pass phrase and R in running in non-interactive mode.

```
library(secret)
try(local_key(), silent = TRUE)
```

```
## Please enter private key passphrase:
```

Create a vault:

You can create a vault by using `create_vault()`

```
vault <- file.path(tempdir(), ".vault")
dir.create(vault)
create_vault(vault)
```

A vault consists of two folders for:

- `users` : contains user and their public keys
- `secrets` : contains the encrypted secrets

```
dir(vault)
```

```
## [1] "README"  "secrets" "users"
```

Alternatively, you can create a vault in an R package:

```
pkg_root <- "/path/to/package"
create_package_vault(pkg_root)
```

Add users to the vault:

To add a user to the vault, you have to know their public key.

The `secret` package contains some public and private keys you can use for demonstration purposes.

```
key_dir <- file.path(system.file(package = "secret"), "user_keys")
alice_public_key <- file.path(key_dir, "alice.pub")
alice_private_key <- file.path(key_dir, "alice.pem")
openssl::read_pubkey(alice_public_key)
```

```
## [2048-bit rsa public key]
## md5: 1d858d316afb8b7d0efd69ec85dc7174
```

Add the public key of Alice to the vault:

```
add_user("alice", alice_public_key, vault = vault)
```

Add a secret using your public key.

A secret can be any R object - this object will be serialised and then encrypted to the vault.

```
secret_to_keep <- c(password = "my_password")
add_secret("secret_one", secret_to_keep, users = "alice", vault = vault)
```

Decrypt a secret by providing your private key:

You can decrypt a secret if you have the private key that corresponds to the public key that was used to encrypt the secret,

```
get_secret("secret_one", key = alice_private_key, vault = vault)
```

```
##      password
## "my_password"
```

Note for Windows users

- If you use windows, you most likely created your keys using PuttyGen. Note that the key created by default from PuttyGen is not in OpenSSH format, so you have to convert your format first. To do this, use the `/Conversions/Export OpenSSH key` menu item in PuttyGen.
- Note that the folder `~/.ssh` in Windows usually expands to `C:\\Users\\YOURNAME\\Documents\\.ssh` . You can find the full path by using:

```
normalizePath("~/.ssh", mustWork = FALSE)
```

```
## [1] "/Users/gaborcsardi/.ssh"
```