pam_sotp 0.3.3 manual

Pedro Diaz <sotp (AT) cavecanen.org>

1. Introduction

pam_sotp provides simple one time password support to PAM, the pluggable authentication modules. This module only provides PAM auth services

1.1. Usage scenario

The user is provided with a list of one time passwords. each time the user wants to authenticate against an sotp-enabled service he or she will be asked to enter an specific password of the list. in case of successful authentication that password will become invalid or valid only for a configured period of time (depending on how the service was configured) and the service will ask for a different password next time.

This list of one time passwords (OTP list) is typically stored in print form and stored in the user's wallet or in electronic form in the user's PDA or cell phone.

Optionally passwords can be configured with a 'prefix', which is another password which the user has to memorize and which has to be written as a prefix of the requested one time password. The prefix acts as a security measure in case that the OTP list is lost or stolen from the user.

The system administrator can configure a service in such a way that entered passwords are valid for a configured period of time. This feature, which we will call password lifespan, is useful when the application requesting the authentication will have to authenticate several more times in a period of time (for example, a webmail application authenticating against the IMAP server).

2. Installing pam_sotp

Starting with version 0.2 pam_sotp has a autoconf-style build system. This means that you should be able to compile & install pam_sotp with the commands:

```
./configure
make
make install
```

As well as the usual arguments (`--prefix`, etc..) the **configure** scripts accepts the following pam_sotp specific arguments:

- `--with-randondev` Device used to gather random data. The default is `/dev/random`. You might want to use the pseudorandom generator `/dev/urandom` instead; password generation will be much faster at the cost of less (cryptographically) strong passwords. If the device given does not exists or is not a character device, glibc's `random()` will be used instead.
- `--with-authdir` Default authentication directory. If not given, `/etc/sotp` will be used
- `--enable-debug` Verbose logging with syslog

pam_sotp requires a `shadow` system group to work. This group will own all the authentication databases and directories in the system. Most linux installations already have this group, so you don't have to worry about this. If you don't have a `shadow` group, (`configure` script will complain about it) try to create it by hand. Also contact me so I can figure out some fix for the next release.

3. Configuration

3.1. Creating a new authentication database

pam_sotp stores authentication information in *authentication databases*. Each authentication database contains the information used to authenticate one user. Authentication databases are grouped into authentication directories. Each pam_sotp-enabled PAM service will authenticate against an authentication directory (by default `/etc/sotp`, but this can be changed for each service). If the user to be authenticated doesn't have an authentication database under the inspected authentication directory, pam_sotp refuses to authenticate him/her, giving control back to the PAM module stack.

The support utilities for pam_sotp have been merged into one utility: **otppasswd**. Once you have installed pam_sotp all you have to do to create a new authentication databasefor your user is to run the **otppasswd** command. **otppasswd** usage is straightforward:

```
$ otppasswd -h
otppasswd v.0.3.3  (C) 2004 Pedro Diaz (sotp@cavecanen.org)

Usage: otppasswd [OPTIONS]

Available options:

  -o file           File used to store the OTP list
  -n number         Number of passwords to generate (default: 20)
  -p prefix         Prefix to add in each generated password (default: No prefix)
  -l length         Length of each generated password (default: 5)
  -t lifespan       Built-in password lifespan, in seconds (default: 0)
  -e days           Make the auth database expire in x days (default: don't expire)
```

```
-c charset        Charset used when generating passwords (default: 0123456789)
-d authdir        Authentication directory (default: /etc/sotp/)
-P                Pretty-print the OTP list
-D                Disable the auth database
-E                Enable the auth database
-h                Show this help message
```

If the options **-D** or **-E** are not specified, **otppasswd** will create a new authentication database overwriting any previous database that the user might have in the authentication directory. This means that you can use **otppasswd** to generate a new OTP list when you are near to run out of passwords. Keep in mind that since **otppasswd** overwrites the previous database your previous OTP list won't be valid anymore.

The option **-D** disables an existing authentication database. The authentication data will still be there but pam_sotp will refuse to use it. The option **-E** enables a previously disabled database.

3.2. Configuring the pam_sotp module

The `pam_sotp.so` module accepts the following options:

- `auth_dir=`*`path`* Path to the SOTP authentication directory. If not specified the default (specified with the **configure** script when pam_sotp was compiled, or `/etc/sotp` if none was specified) will be used
- `fail_delay=`*`nsecs`* Plan a delay of at least *`nsecs`* seconds after a failed authentication. The actual number of seconds of waiting depends on PAM and the module stack. The default for this option is zero seconds.
- `prompt_number=`*<`yes`|`no`>* Include the password number in the prompt. The default for this option is to include the password number in the prompt.
- `pw_lifespan=`*`nsecs`* Set the password lifespan value for this service. The default for this option is cero seconds.

3.2.1. Example

We want to configure the login service with SOTP support. We also want to let in other non-SOTP users. SOTP users must also be able to log-in with their regular password.

The file `/etc/pam.d/login` will look something like this:

```
auth sufficient pam_sotp.so prompt_number=yes
account required        pam_unix.so
```

SOTP users (the ones with an auth database in `/etc/sotp`) will be asked for an OTP. If the OTP is wrong they will be asked for their regular password:

```
odiel:~# telnet localhost
Trying 127.0.0.1...
Connected to localhost.localdomain.
Escape character is '^]'.
Debian GNU/Linux 3.1 odiel
odiel login: susan

One time password [01]: ❶
Password:
```

❶ The wrong OTP was introduced

Non-SOTP users won't notice difference:

```
odiel:~# telnet localhost
Trying 127.0.0.1...

Connected to localhost.localdomain.
Escape character is '^]'.
Debian GNU/Linux 3.1 odiel
odiel login: pdiaz
Password:
```

3.3. Other issues

3.3.1. Password lifespan

With the options `pw_lifespan` in the module configuration and `-t` in **otppasswd** you can configure pam_sotp to authenticate a service with *limited time passwords*. This basically means that pam_sotp will handle authentication in the same way as before but with the following additional feature: an used password will remain valid for some time after its first use. Of course, the prompted password will be also valid.

The actual amount of time an older password will remain valid is calculated as the minimum of the value specified in `/etc/pam.d/service` configuration file and the value specified with **otppasswd** when the user database was created. This means that specifying a value of zero in either of these places will disable password lifespan for that pam service or authentication database.

Limited Time Passwords are useful for using pam_sotp with software that requires several authentications against a service in a short period of time, such as some webmail systems (Squirrelmail (http://www.squirrelmail.org/) has this behavior).

4. Disclaimer, contact information, etc...

pam_sotp is *not* yet ready for production use. This is alpha-quality code, it may be full of remote holes (I hope not ;-). Having said that, I've been using pam_sotp for accessing my webmail for some time and I didn't found any problems.

This is my first PAM module. I've done my best to comply with the standards recommended by the PAM Module Writers' Manual. If you have experience writing PAM modules and you have any suggestions about the code, please contact me!

Suggestions, patches and code contributions can be directed to <`sotp (AT) cavecanen.org`>. They will be very welcome.