

UniOTP PAM Authentication Agent User Guide

Contents

UniOTP PAM Authentication Agent User Guide.....	1
1 About This Document	3
2 PAM Authentication Agent installation and configuration	3
2.1 Installation.....	3
2.1.1 Extract pam_secu.tar.gz File	3
2.1.2 Install PAM Authentication Agent	4
2.1.3 Check the PAM Agent Installation	5
2.2 PAM Agent Configuration	6
2.2.1 Configure the secu_pam_agent.conf file.....	6
2.2.2 Configure the PAM Authentication Agent	7
2.2.3 Configure command line login authentication	7
2.2.4 Configure Desktop login authentication.....	9
2.2.5 Configure remote terminal login authentication	9
3 Testing PAM Authentication Agent.....	10
3.1 Testing Command line authenticaiton	10
3.2 Testing Desktop login authentication	11
3.3 Testing remote login authentication	12

1 About This Document

This User guide includes introduction of UniOTP PAM Authentication Agent installation and configuration in Red Hat Enterprise Linux 5 system, and using the UniOTP authentication system to protect login authentication.

Environment Used

1. Operating system: Red Hat Enterprise 5

2 PAM Authentication Agent installation and configuration

2.1 Installation

2.1.1 Extract pam_secu.tar.gz File

Go to the directory where you save your setup package (pam_secu.tar.gz). For example, in our system, the setup package is stored in /tmp/Desktop. Use cd command to switch to that folder, and tar command to extract the setup package, as the following picture.

```
[root@localhost tmp]# cd Desktop
[root@localhost Desktop]# ls
pam_secu.tar.gz
[root@localhost Desktop]# tar zxvf pam_secu.tar.gz
```

You will see the following information, after the package is extracted. All extracted files will be stored in the same directory as the pam_secure.tar.gz.

```
[root@localhost Desktop]# tar zxvf pam_secu.tar.gz
./libc.so.6
./libstdc++.so.5
./libuniotp_agent_c.so
./pam_secu.so
./secu_pam_agent.conf
./install
[root@localhost Desktop]# _
```

2.1.2 Install PAM Authentication Agent

To install the PAM Agent, firstly, you must login as root and then input “./install” and press enter to start install PAM Agent.

```
[root@localhost Desktop]# ./install
```

The following information will display.

```
Welcome to use UniOTP PAM Agent
Checking dependent libraries...
Checking libstdc++.so.5...
Library libstdc++.so.5 [0k]
Checking libc.so.6...
Library libc.so.6 [0k]
Install libuniotp_agent_c.so ...
Install sample configuration file ...
pam_secu.so has been installed in /lib/security/
Change configuration file now?[y/n]
```

After installation, the system will ask you “Change configuration file now? [y/n]”. If you want to change the configuration file now, input y and press enter, otherwise, input n and press enter. Here, input n and press enter, and I will change the configuration file later. The following information will display.



Please make sure the configuration file has been configured correctly, before restart your computer, otherwise it may cause login failure next time.

```

ñ
UniOTP PAM Agent installation has been finished.
Configuration file(secu_pam_agent.conf) for UniOTP PAM Agent
is installed /etc/
Before you restart your computer after you configure an
application to use UniOTP One-Time-Password to do authentication,
you should confirm you have configure the
configuration file(/etc/secu_pam_agent.conf) correctly!

```

2.1.3 Check the PAM Agent Installation

You can check if the PAM agent has been installed successfully, by the following method.

1. Go to /lib/security to check the pam_secu.so file.

```

[root@localhost security]# ls /lib/security
pam_access.so      pam_keyinit.so    pam_permit.so      pam_tally2.so
pam_ccreds.so      pam_krb5           pam_pkcs11.so      pam_tally.so
pam_chroot.so      pam_krb5afs.so    pam_postgresok.so  pam_time.so
pam_console.so     pam_krb5.so       pam_pwhistory.so   pam_timestamp.so
pam_cracklib.so    pam_lastlog.so    pam_rhosts_auth.so pam_tty_audit.so
pam_debug.so       pam_ldap.so       pam_rhosts.so      pam_umask.so
pam_deny.so        pam_limits.so     pam_rootok.so      pam_unix_acct.so
pam_echo.so        pam_listfile.so   pam_rps.so         pam_unix_auth.so
pam_env.so         pam_localuser.so  pam_securetty.so   pam_unix_passwd.so
pam_exec.so        pam_loginuid.so   pam_secu.so        pam_unix_session.so
pam_faildelay.so   pam_mail.so       pam_selinux.so     pam_unix.so
pam_filter         pam_mkhome.so     pam_shells.so      pam_userdb.so
pam_filter.so      pam_motd.so       pam_smb_auth.so    pam_warn.so
pam_ftp.so         pam_namespace.so  pam_stack.so       pam_wheel.so
pam_group.so       pam_nologin.so    pam_stress.so      pam_xauth.so
pam_issue.so       pam_passwdqc.so   pam_succeed_if.so
[root@localhost security]#

```

2. Go to /user/lib to check the libuniotp_agent_c.so file.

```

libtiff.so.3
libtiff.so.3.8.2
libtiffxx.so.3
libtiffxx.so.3.8.2
libtk8.4.so
libuniotp_agent_c.so

```

3. Go to /etc to check the secu_pam_agent.conf file

```

[root@localhost etc]# ls /etc/secu_pam_agent.conf
/etc/secu_pam_agent.conf

```

If all these files can be found in the corresponding directory, the PAM Agent has been installed successfully, and the next step is to configure the PAM Agent.

2.2 PAM Agent Configuration

2.2.1 Configure the secu_pam_agent.conf file

This file contains settings about system account, mapping of dynamic password name for system account, authentication server IP address shared key and authentication server port. For more details, please read introduction in the secu_pam_agent.conf file.

```
[root@localhost etc]# vi secu_pam_agent.conf
```

The following file will be opened. Add a system user and other corresponding parameters as following picture.

```
[root]
uniotp_account=newtest
authserver=192.168.1.225
share=hello
port=1812
maxwait=3
```



[User name]: a valid system account name (example: [root])

uniotp_account: the mapping account name of dynamic password for [user name]

authserver: authentication server IP address

share: shared secret key (obtained from authentication service administrator)

port: authentication server connecting port

maxwait: the maximum time waiting for authentication server response

2.2.2 Configure the PAM Authentication Agent

The PAM authentication agent configuration is that adding the OTP Server PAM authentication agent program to the authentication module service, which is in /etc/pam.d.

```
[root@localhost pam.d]# cd /etc/pam.d
```

2.2.3 Configure command line login authentication

Open the login file in directory “/etc/pam.d”, and add “auth required pam_secu.so” in the row.

```
[root@localhost pam.d]# vi /etc/pam.d/login
```

```
#%PAM-1.0
auth      required      /lib/security/pam_secu.so
auth      user_unknown=ignore success=ok ignore=ignore default=badl pam_securetty.so
auth      include       system-auth
account   required      pam_nologin.so
account   include       system-auth
password  include       system-auth
```

Save and quit. Now you have finished all PAM agent configurations for command line login authentication.



There are four control-flag keywords: required, requisite, sufficient, optional and include.

required; this indicates that the success of the module is required for the module-type facility to succeed. Failure of this module will not be apparent to the user until all of the remaining modules (of the same module-type) have

been executed.

requisite; like required, however, in the case that such a module returns a failure, control is directly returned to the application. The return value is that associated with the first required or requisite module to fail. Note, this flag can be used to protect against the possibility of a user getting the opportunity to enter a password over an unsafe medium. It is conceivable that such behavior might inform an attacker of valid accounts on a system. This possibility should be weighed against the not insignificant concerns of exposing a sensitive password in a hostile environment.

sufficient; the success of this module is deemed 'sufficient' to satisfy the Linux-PAM library that this module-type has succeeded in its purpose. In the event that no previous required module has failed, no more 'stacked' modules of this type are invoked. (Note, in this case subsequent required modules are not invoked.). A failure of this module is not deemed as fatal to satisfying the application that this module-type has succeeded.

optional; as its name suggests, this control-flag marks the module as not being critical to the success or failure of the user's application for service. In general, Linux-PAM ignores such a module when determining if the module stack will succeed or fail. However, in the absence of any definite successes or failures of previous or subsequent stacked modules this module will determine the nature of the response to the application. One example of this latter case, is when the other modules return something like PAM_IGNORE.

include; this tells PAM to include all lines of given type from the configuration file specified as an argument to this control. The whole idea is to create few "systemwide" pam configs and include parts of them in application pam configs.



pam_secu.so support debug function. If you input debug following pam_secu.so the debug information will be added to the system log file (like: auth sufficient pam_secu.so debug).

And you can specify the configuration file by using `conf -directory` (like: `auth sufficient pam_secu.so conf -/etc/secu_pam_agent.conf`, the agent will find configuration file from the specified directory, instead of the default one).

2.2.4 Configure Desktop login authentication

Open the PAM configuration file for gdm in directory `/etc/pam.d`, and add `auth required pam_secu.so` in the first row.

```
[root@localhost ~]# vi /etc/pam.d/gdm_
```

```
#%PAM-1.0
auth      sufficient      pam_secu.so
auth      required       pam_env.so
auth      include         system-auth
```

Save and quit. Now you have finished all PAM agent configurations for desktop login authentication.



Please make sure the configuration file has been configured correctly, before restart your computer, otherwise it may cause login failure next time.

2.2.5 Configure remote terminal login authentication

Open the `sshd` file in directory `/etc/pam.d` and add `auth sufficient pam_secu.so sshd` in the row.

```
[root@localhost pam.d]# vi sshd
```

```
#%PAM-1.0
auth      sufficient    pam_secure.so          sshd
auth      include      system-auth
account   required      pam_nologin.so
account   include      system-auth
password  include      system-auth
session   optional      pam_keyinit.so force revoke
session   include      system-auth
session   required      pam_loginuid.so
```



For remote login authentication configuration, you can only use sufficient.

Save and quit. Now you have finished all PAM agent configurations for remote terminal login authentication.

3 Testing PAM Authentication Agent

3.1 Testing Command line authentication

Input user name [root] and press enter. Now the system will ask for OTP[PIN]:. Just input the dynamic password generated by your OTP device and your PIN following OTP in this line and press enter. In the next Password line, input your static password and press enter. You will login as root, if you see [root@localhost ~]#.

```
Red Hat Enterprise Linux Server release 5.6 (Tikanga)
Kernel 2.6.18-238.el5 on an i686

localhost login: root
OTP[PIN]:
Password:
Last login: Thu Jul 14 00:38:50 on tty2
[root@localhost ~]#
```

3.2 Testing Desktop login authentication

After configuring the gdm file, you can login the system by using the GUI, input the user name when the following login interface appear.



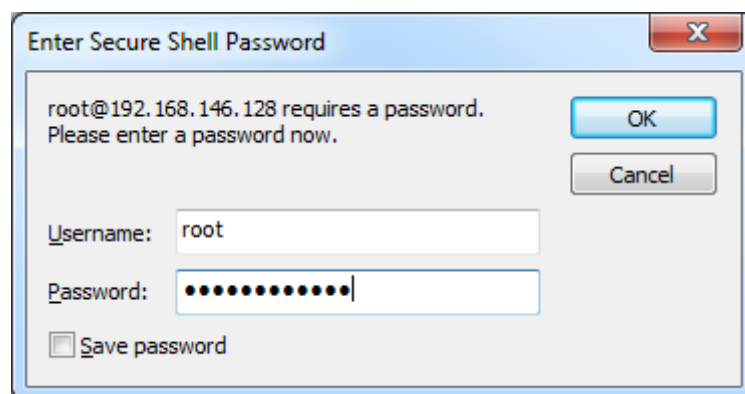
Input user name [root], and then the password interface will appear.



Input OTP and PIN, and then press enter. If you can login the system, you have successfully configure the PAM agent for desktop login authentication.

3.3 Testing remote login authentication

We use SecureCRT as the remote login tools. After UniOTP protection is enabled for remote login authentication. You must input OTP+PIN in password to login the host computer, and click on OK.



The login has been successfully, as the following picture.

