SaltStack extension for SAP PSE

Release 1.0.0

'Benjamin Wegener, Alexander Wilke'

Nov 24, 2022

CONTENTS:

CHAPTER

ONE

COMPLETE LIST OF SAP_PSE

1.1 Execution Modules

saltext.sap_pse._modules.sap_pse	SaltStack extension for sapgenpse Copyright (C) 2022 SAP UCC Magdeburg

1.1.1 saltext.sap_pse._modules.sap_pse

SaltStack extension for sapgenpse Copyright (C) 2022 SAP UCC Magdeburg

sapgenpse execution module

SaltStack execution module that wraps sapgenpse functions.

codeauthor
Benjamin Wegener, Alexander Wilke

maturity
new

depends
yaml

platform
Linux

It is assumed that the program `sapgenpse` is in the `PATH` of the user executing the function. If not, it is assumed that the SAP Host Agent is installed and that `/usr/sap/hostctrl/exe/sapgenpse` can be acessed by the executing user. If the executing user is not provided, the user under which the salt minion runs is used (usually `root`).

`saltext.sap_pse._modules.sap_pse.__virtual__()`
Only work on POSIX-like systems

`saltext.sap_pse._modules.sap_pse.gen_pse(`*pse_file*, *dn*, *pse_pwd=None*, *algo='RSA:2048:SHA512'*, *runas=None*, *groupas=None*, *add_ca_bundle=True*, ***kwargs*`)`

Wrapper for the function `gen_pse` of the CLI tool `sapgenpse`.

Create a new PSE. This will **not** create a signing request.

pse_file
Equivalent to `-p <pse-file>`, i.e. the path for the PSE.

dn
Distinguished name.

pse_pwd
Equivalent to `-x <pin>`, i.e. the PIN/Passphrase for the PSE. Default is no PIN.

algo
Equivalent to `-a <algo>`, i.e. the algorithm used for the PSE, e.g. DSA, ECDSA or RSA (default is `RSA:2048:SHA512`).

runas
User that will run the command, default is the user that runs the salt minion.

groupas
Group that will run the command, default is the group that runs the salt minion.

add_ca_bundle
If False, will not add the OpenSSL CA bundle returned by `salt.utils.http.get_ca_bundle()` which is all certificate authorities that are trusted by the operating system.

Returns True / False based on success.

CLI Example:

```
salt "*" sap_pse.gen_pse pse_file="/usr/sap/hostctrl/exe/sec/SAPSSLA.pse" dn=
↪"cn=ANONYMOUS"
```

saltext.sap_pse._modules.sap_pse.**import_p8**(*pse_file*, *pub_key*, *priv_key*, *pse_pwd=None*, *priv_key_pwd=None*, *add_certs=None*, *runas=None*, *groupas=None*, *add_ca_bundle=True*, ***kwargs*)

Wrapper for the function `import_p8` of the CLI tool `sapgenpse`.

This function creates a new PSE file from a PKCS#8 format private key (optionally protected by PKCS#5 password-based encryption) along with all necessary X.509 certs.

You will have to supply the X.509 certificate matching the private key plus all intermediate and root CA certificates which might be necessary to build a certificate chain that ends with a self-signed certificate.

pse_file
Equivalent to `-p <pse-file>`, i.e. the path for the PSE.

pse_pwd
Equivalent to `-x <pin>`, i.e. the PIN/Passphrase for the PSE. Default is no PIN.

pub_key
Equivalent to `-c <cert(s)-file>`, i.e. a X.509 certificate containing the public key.

priv_key
Path to the X.509 certificate containing the private key.

priv_key_pwd
Equivalent to `-z <password>`, i.e. the Password/Passphrase for decryption of private key. Default is no password.

add_certs
Equivalent to `-r <file2>`, i.e. additional certificate(s) for an incomplete PKCS#8 file. This list can contain to 10 additional files for building complete certification path up to the RootCA (PEM, Base64 or DER binary). Default is no additional certificates.

runas
User that will run the command, default is the user that runs the salt minion.

groupas
Group that will run the command, default is the group that runs the salt minion.

add_ca_bundle
If False, will not add the OpenSSL CA bundle returned by `salt.utils.http.get_ca_bundle()` which is all certificate authorities that are trusted by the operating system.

Returns True / False based on success.

CLI Example:

```
salt "*" sap_pse.import_p8 pse_file="/usr/sap/hostctrl/exe/sec/SAPSSLS.pse" pub_key=
↪"/etc/pki/cert.crt" priv_key="/etc/pki/cert.key"
```

saltext.sap_pse._modules.sap_pse.**export_p8**(*pse_file*, *pem_file*, *pem_pwd*, *pse_pwd=None*, *runas=None*, *groupas=None*, *secudir=None*, ***kwargs*)

Wrapper for the function `export_p8` of the CLI tool `sapgenpse`.

Exports the key of a PSE into PKCS#8 transfer format (PEM-File) for transfer/export to software of other vendors.

The private key and its corresponding certificat plus forward certificate chain up to and including the RootCA's certificate are written into a PEM file.

pse_file
Equivalent to `-p <pse-file>`, i.e. the path of the PSE.

pem_file
Path to the PEM file which will contain both public and private key.

pem_pwd
Equivalent to `-z <password>`, i.e. the Password/Passphrase for the encryption of the PEM-file.

pse_pwd
Equivalent to `-x <pin>`, i.e. the PIN/Passphrase for PSE file. Default is no PIN.

runas
User that will run the command, default is the user that runs the salt minion.

groupas
Group that will run the command, default is the group that runs the salt minion.

secudir
SECUDIR to use. If not defined, the path of the PSE file will be set as SECUDIR.

Returns True / False based on success.

CLI Example:

```
salt "*" sap_pse.export_p8 pse_file="/usr/sap/hostctrl/exe/sec/SAPSSLS.pse" pem_
↪file="/etc/pki/pse.crt" pem_pwd=Abcd1234
```

saltext.sap_pse._modules.sap_pse.**get_my_name**(*pse_file*, *pse_pwd=None*, *runas=None*, *groupas=None*, *secudir=None*, ***kwargs*)

Wrapper for the function `get_my_name` of the CLI tool `sapgenpse`.

Displays the attributes/properties of the user/owner certificate in a PSE.

pse_file
Equivalent to `-p <pse-file>`, i.e. the path of the PSE.

pse_pwd
Equivalent to `-x <pin>`, i.e. the PIN/Passphrase for PSE file. Default is no PIN.

runas
User that will run the command, default is the user that runs the salt minion.

groupas
Group that will run the command, default is the group that runs the salt minion.

secudir
SECUDIR to use. If not defined, the path of the PSE file will be set as SECUDIR.

CLI Example:

```
salt "*" sap_pse.get_my_name pse_file="/usr/sap/hostctrl/exe/sec/SAPSSLS.pse"
```

saltext.sap_pse._modules.sap_pse.**maintain_pk_add**(*pse_file*, *certs*, *runas=None*, *groupas=None*, *pse_pwd=None*, *secudir=None*, ***kwargs*)

Wrapper for the function `maintain_pk` of the CLI tool `sapgenpse`.

Adds certificates to the PK list of a PSE.

pse_file
Equivalent to `-p <pse-file>`, i.e. the path of the PSE.

certs
Equivalent to `-m <cert-file>`, i.e. add multiple certificates from <file>. Must be a list.

pse_pwd
Equivalent to `-x <pin>`, i.e. the PIN/Passphrase for PSE file. Default is no PIN.

runas
User that will run the command, default is the user that runs the salt minion.

groupas
Group that will run the command, default is the group that runs the salt minion.

secudir
SECUDIR to use. If not defined, the path of the PSE file will be set as SECUDIR.

CLI Example:

```
salt "*" sap_pse.maintain_pk_add pse_file="/usr/sap/hostctrl/exe/sec/SAPSSLS.pse"␣
↪certs=["/etc/pki/trust/anchors/ca.crt"]
```

saltext.sap_pse._modules.sap_pse.**maintain_pk_delete**(*pse_file*, *del_cert*, *runas=None*, *groupas=None*, *pse_pwd=None*, *secudir=None*, ***kwargs*)

Wrapper for the function `maintain_pk` of the CLI tool `sapgenpse`.

Delete certificates from the PKList of a PSE.

pse_file
Equivalent to `-p <pse-file>`, i.e. the path of the PSE.

del_cert
Equivalent to "`-d <num>` (delete certificate/key number <num> from PKList) or `-d <string>` (delete certificates/keys from PKList containing <string>)

pse_pwd
Equivalent to `-x <pin>`, i.e. the PIN/Passphrase for PSE file. Default is no PIN.

runas
User that will run the command, default is the user that runs the salt minion.

groupas
Group that will run the command, default is the group that runs the salt minion.

secudir
SECUDIR to use. If not defined, the path of the PSE file will be set as SECUDIR.

CLI Example:

```
salt "*" sap_pse.maintain_pk_delete pse_file="/usr/sap/hostctrl/exe/sec/SAPSSLS.pse
↪" del_cert=0
```

saltext.sap_pse._modules.sap_pse.**maintain_pk_list**(*pse_file*, *runas=None*, *groupas=None*, *pse_pwd=None*, *secudir=None*, ***kwargs*)

Wrapper for the function `maintain_pk` of the CLI tool `sapgenpse`.

List certificates from the PKList of a PSE.

pse_file
Equivalent to `-p <pse-file>`, i.e. the path of the PSE.

pse_pwd
Equivalent to `-x <pin>`, i.e. the PIN/Passphrase for PSE file. Default is no PIN.

runas
User that will run the command, default is the user that runs the salt minion.

groupas
Group that will run the command, default is the group that runs the salt minion.

secudir
SECUDIR to use. If not defined, the path of the PSE file will be set as SECUDIR.

CLI Example:

```
salt "*" sap_pse.maintain_pk_list pse_file="/usr/sap/hostctrl/exe/sec/SAPSSLS.pse"
```

saltext.sap_pse._modules.sap_pse.**seclogin_add**(*pse_file*, *pse_pwd=None*, *user=None*, *runas=None*, *groupas=None*, *secudir=None*, ***kwargs*)

Wrapper for the function `seclogin` of the CLI tool `sapgenpse`.

Creates Single Sign-On (SSO) credentials for a PSE / user.

pse_file
Equivalent to `-p <pse-file>`, i.e. the path of the PSE.

pse_pwd
Equivalent to `-x <pin>`, i.e. the PIN/Passphrase for PSE file. Default is no PIN.

user
Equivalent to `-O <username>`, i.e. create SSO-credential for OTHER user <username>. Will be set to runas or salt minion user if None.

runas
User that will run the command, default is the user that runs the salt minion.

groupas
Group that will run the command, default is the group that runs the salt minion.

secudir
SECUDIR to use. If not defined, the path of the PSE file will be set as SECUDIR.

CLI Example:

```
salt "*" sap_pse.seclogin_add pse_file="/usr/sap/hostctrl/exe/sec/SAPSSLS.pse" user=
↪"sapadm"
```

saltext.sap_pse._modules.sap_pse.**seclogin_contains**(*pse_file*, *pse_pwd=None*, *user=None*, *runas=None*, *groupas=None*, *secudir=None*, ***kwargs*)

Wrapper for the function `seclogin` of the CLI tool `sapgenpse`.

Returns success and if Single Sign-On (SSO) credentials for user already exist.

pse_file
Equivalent to `-p <pse-file>`, i.e. the path of the PSE.

pse_pwd
Equivalent to `-x <pin>`, i.e. the PIN/Passphrase for PSE file. Default is no PIN.

user
Equivalent to `-O <username>`, i.e. create SSO-credential for OTHER user <username>. Will be set to runas or salt minion user if None.

runas
User that will run the command, default is the user that runs the salt minion.

groupas
Group that will run the command, default is the group that runs the salt minion.

secudir
SECUDIR to use. If not defined, the path of the PSE file will be set as SECUDIR.

CLI Example:

```
salt "*" sap_pse.seclogin_contains pse_file="/usr/sap/hostctrl/exe/sec/SAPSSLS.pse"␣
↪user="sapadm"
```

saltext.sap_pse._modules.sap_pse.**seclogin_count**(*pse_file*, *runas=None*, *groupas=None*, *pse_pwd=None*, *secudir=None*, ***kwargs*)

Wrapper for the function `seclogin` of the CLI tool `sapgenpse`.

Returns success and the count of SSO credentials for the given PSE.

pse_file
Equivalent to `-p <pse-file>`, i.e. the path of the PSE.

pse_pwd
Equivalent to `-x <pin>`, i.e. the PIN/Passphrase for PSE file. Default is no PIN.

runas
User that will run the command, default is the user that runs the salt minion.

groupas
Group that will run the command, default is the group that runs the salt minion.

secudir
SECUDIR to use. If not defined, the path of the PSE file will be set as SECUDIR.

CLI Example:

```
salt "*" sap_pse.seclogin_count pse_file="/usr/sap/hostctrl/exe/sec/SAPSSLS.pse"
```

saltext.sap_pse._modules.sap_pse.**seclogin_delete**(*pse_file*, *pse_pwd=None*, *runas=None*, *groupas=None*, *secudir=None*, ***kwargs*)

Wrapper for the function `seclogin` of the CLI tool `sapgenpse`.

Removes all SSO credentials for a PSE file.

pse_file
Equivalent to `-p <pse-file>`, i.e. the path of the PSE.

pse_pwd
Equivalent to `-x <pin>`, i.e. the PIN/Passphrase for PSE file. Default is no PIN.

runas
User that will run the command, default is the user that runs the salt minion.

groupas
Group that will run the command, default is the group that runs the salt minion.

secudir
SECUDIR to use. If not defined, the path of the PSE file will be set as SECUDIR.

CLI Example:

```
salt "*" sap_pse.seclogin_delete pse_file="/usr/sap/hostctrl/exe/sec/SAPSSLS.pse"
```

saltext.sap_pse._modules.sap_pse.**gen_verify_pse**(*pse_file=None*, *runas=None*, *groupas=None*, ***kwargs*)

Wrapper for the function `gen_verify_pse` of the CLI tool `sapgenpse`.

Create a new PSE for verification without own key pair.

pse_file
Equivalent to `-p <pse-file>`, i.e. the path of the PSE.

runas
User that will run the command, default is the user that runs the salt minion.

groupas
Group that will run the command, default is the group that runs the salt minion.

Note: This will utilze the OpenSSL CA bundle returned by `salt.utils.http.get_ca_bundle()`.

CLI Example:

```
salt "*" sap_pse.seclogin_delete pse_file="/usr/sap/hostctrl/exe/sec/SAPSSLS.pse"
```

1.2 State Modules

saltext.sap_pse._states.sap_pse	SaltStack extension for sapgenpse Copyright (C) 2022 SAP UCC Magdeburg

1.2.1 saltext.sap_pse._states.sap_pse

SaltStack extension for sapgenpse Copyright (C) 2022 SAP UCC Magdeburg

sapgenpse state module

SaltStack module that implements states based on sapgenpse functionality.

codeauthor
Benjamin Wegener, Alexander Wilke

maturity
new

depends
N/A

platform
Linux

This module implements states that utilize sapgenpse functionality and manages SAP PSEs (Personal Security Environment).

Note: This module can only run on linux platforms.

---

saltext.sap_pse._states.sap_pse.**managed**(*name*, *user=None*, *group=None*, *seclogons=None*, *pin=None*, *priv_key=None*, *priv_key_pw=None*, *pub_key=None*, *trusted_certs=None*, *backup=False*, *add_ca_bundle=True*, *dn=None*, ***kwargs*)

Create or manage a SAP PSE keystore based on a public / private key pair. If not public / private key pair is given, a PSE with the given DN is managed.

name
The path to the pse file.

user
User to run all commands, e.g. sidadm. If not provided, will default to the either the owner of the PSE file or to user that runs the salt minion.

group
Group under which all commands are run.

seclogons:
List of users to store SSO credentials for. Empyty by default.

pin
The pin of the keystore.

priv_key
Private key file, e.g. be `/etc/pki/{{ __grains__["id"] }}.key`

priv_key_pw
Private key password, default is None

pub_key
Public key file, e.g. be `/etc/pki/{{ __grains__["id"] }}.crt`

trusted_certs
List of trusted certificates that should be added to the PSE.

backup
Set to True if a backup of an existing file should be made.

add_ca_bundle
Set to False if the VMs CA bundle should **not** be added to the PSE during creation.

dn
Distinguished Name of the PSE.

The intended use of this state is to take a previously signed X.509 keypair and create a PSE based on the these files. The PSE can then be consumed by other applications (e.g. Host Agent, HANA, NetWeaver etc.).

Note: Remember to inform the application of changes to the PSE (re-/created)!

Example:

```
SAP Host Agent PSE is managed:
  sap_pse.managed:
    - name: /usr/sap/hostctrl/exe/sec/SAPSSLS.pse
    - user: sapadm
    - group: sapsys
    - seclogons:
      - sapadm
    - pin: __slot__:salt:vault.read_secret(path="certstores/pse", key="/usr/sap/
↪hostctrl/exe/sec/SAPSSLS.pse")
    - priv_key: /etc/pki/{{ grains["id"] }}.key
    - pub_key: /etc/pki/{{ grains["id"] }}.crt
    - backup: True
```

saltext.sap_pse._states.sap_pse.**absent**(*name*, *secudir=None*, *user=None*, *pin=None*, ***kwargs*)

Ensure that a PSE is absent from the system.

name
Name of the PSE file.

secudir
SECUDIR variable, required to determine location of cred_v2 SSO credential files.

user
User to run the command with.

pin
The pin of the keystore.

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

S

INDEX

Symbols

A

E

G

I

M

S