PAM Ansible Integration

Credential in Ansible environment might be needed at 3 different occasions.

1. Ansible tower server compile time. Ansible server compile the ansible code and the send the target server to run the Ansible Task. Ansible need credential during this time. For this we can develop a lookup plugin called 'a2apassword'
2. Ansible hosts are managed through PAM and need root account credential to run Ansible play book. For this we update ssh connection plugin. New plugin we name it pamplugin.
3. Compiled Ansible code running on target hosts need to retrieve credential. Use normal a2a coding for this purpose and this document does not cover this use case.

Lookup Plugin

Plugin Setup

Copy below code a2apassword.py into /usr/share/ansible/plugins/lookup/a2apassword.py

a2apasswd.py

```
# (c) 2020, Kannan-Chairman <ck_kannan(at)yahoo.com>
# (c) 2017 Ansible Project
# Copy this file to /usr/share/ansible/plugins/lookup/a2apassword.py
from __future__ import (absolute_import, division, print_function)
__metaclass__ = type

DOCUMENTATION = """
    lookup: a2apassword
    author: Kannan-Chairman <ck_kannan(at)yahoo.com>
    version_added: "1.0"
    short_description: Read password from Broadcom PAM using A2A agent
    requirement:
      - A2A Agent installed on Ansible Control Hosts
    description:
      - Read password for a Target Alias from PAM vault
    options:
      targetAlias:
        description:  Name of Targer Alias
      a2adir:
        description: Location of A2A Folder Used to set environment variable CSPM_CLIENT_HOME
        default: /opt/a2a/catech
      bypasscache:
        description: Enable Cache ByPass
        default: False
      _command:
        description: A2A Command to retrive password
        env:
          - name: A2A_CLIPASSWORD_CMD
        default: '/opt/a2a/catech/cspmclient/bin/cspmclient64'
    output:
      description: Output format. txt for password in text. xml to return xml data
      default:  'txt'
    notes:
"""

EXAMPLES = """
  - name: Get Password for alias from A2a
     debug: msg={{ lookup("a2apassword", a2acmd)}}
     vars:
       a2acmd:
         targetAlias: DIST_tsmadmin
         bypasscache: False
         a2adir: /opt/a2a/catech
         output: xml

"""

RETURN = """
      - if output is txt. Return value is 'status  ID   Password'
      - if output is xml returns result in xml
```

```
"""


import os
import subprocess
from subprocess import PIPE
from subprocess import Popen

from ansible.errors import AnsibleError, AnsibleAssertionError
from ansible.plugins.lookup import LookupBase
from ansible.parsing.splitter import parse_kv
from ansible.module_utils._text import to_bytes, to_text, to_native

try:
    from __main__ import display
except ImportError:
    from ansible.utils.display import Display
    display = Display()

A2A_CLIPASSWORD_CMD = os.getenv('A2A_CLIPASSWORD_CMD', '/opt/a2a/catech/cspmclient/bin/cspmclient64')

class A2Apassword:
    """
    Run a2aprocess and retrive password
    """
    def __init__(self, targetAlias=None, bypasscache=False , a2adir='/opt/a2a/catech', output='txt', **kwargs ):
##      self.targetAlias = 'DIST_tsmadmin'
        self.targetAlias = targetAlias
        self.command     = A2A_CLIPASSWORD_CMD
        self.a2adir      = a2adir
        os.putenv('CSPM_CLIENT_HOME',self.a2adir)
        if self.targetAlias is None:
          raise AnsibleError("A2A Password Error: TargetAlias not specified")
        if output == 'txt':
          self.output=' '
        else:
          self.output = '-x'
        if bypasscache is False:
          self.bypasscache = ' '
        else:
          self.bypasscache = '-b'
        display.v("Value of output {} Value of self.output {}".format(output,self.output))

    def get(self):
      try:
        all_params = [A2A_CLIPASSWORD_CMD, self.targetAlias,self.bypasscache, self.output]
        credential=""
        print(all_params)
        tmp_output, tmp_error = Popen(all_params, stdout=PIPE, stderr=PIPE, stdin=PIPE).communicate()
        if tmp_output:
           credential = tmp_output
        if tmp_error:
           raise AnsibleError("ERROR => %s " % (tmp_error))
      except subprocess.CalledProcessError as e:
        raise AnsibleError(e.output)
      except OSError as e:
        raise AnsibleError("ERROR: Cli execution failed ERROR =(%s) => %s " % (to_text(e.errno), e.strerror))
      return credential


class LookupModule(LookupBase):
    """
    USAGE:  lookup(a2apassword,DIST_tsmadmin)
    """
    def run(self, terms,  variables=None, **kwargs ):
      display.vvvv(terms)
      if isinstance(terms, list):
        return_values = []
        for term in terms:
          display.vvvv("Term: %s" % term)
          a2a_conn = A2Apassword(**term)
```

```
          result = a2a_conn.get()
          return_values.append(result)
        return return_values
      else:
        a2a_conn = A2Apassword(**term)
        result = a2a_conn.get()
        return  result
```

a2apassword Usage Example

a2aLookup Example

```
  - hosts: mnncpold51016
    name: A2A test
    vars:
      contents: "{{ lookup('a2apassword',a2ademo) }}"
      a2ademo:
        targetAlias: DIST_testuser1000
        a2adir: '/opt/a2a/catech/'
        bypasscache: True
        output: 'txt'
    tasks:
```

SSH Connection Plugin

Let us say for our discussion the Ansible hosts root accounts are managed in PAM and Ansible connect to target host using 'root' account. i.e. Direct login is enabled. For now do not worry about security issue with direct login. Technically the plugin should work with out direct login however I have not tested it. Before Ansible reaches target hosts Ansible need to retrieve target host credential and remove it after the ansible complete its use. High level idea is add a trigger script in ssh connection which trigger A2A request and setup required trust. Below are the steps to do this.

1. Update ssh connection plugin to add Trigger feature. (i.e. add pamintegration variable when set to yes will call pamscript.
2. Develop the pam trigger which will connect to PAM and setup hosts private keys.
3. Update Ansible to use private key to use hostname.

Update of Setup pamssh connection Plugin

When Ansible connect to target hosts ansible, we can update ansible connection pulgin to add a feature can connect to PAM through a2a client and retrieve credential and setup the We can add Ansible ssh.py connection plugin with 2 options where is pamintegration is needed it will run a pre script to retrieve credentials. I took ssh.py and made this update and call this pamssh.py.

Copy /usr/lib/python2.7/site-packages/ansible/plugins/connection/ssh.py to /usr/share/ansible/plugins/connection/pamssh.py and then patch this file using below command

pamssh patch

```
 patch /usr/share/ansible/plugins/connection/pamssh.py   ssh.patch.pamssh

Content of ssh.patch.pamssh is
--- /usr/lib/python2.7/site-packages/ansible/plugins/connection/ssh.py  2020-04-16 21:30:19.000000000 -0400
+++ /usr/share/ansible/plugins/connection/pamssh.py     2021-07-07 19:47:36.175557575 -0400
@@ -3,12 +3,13 @@
 # Copyright 2017 Toshio Kuratomi <tkuratomi@ansible.com>
 # Copyright (c) 2017 Ansible Project
 # GNU General Public License v3.0+ (see COPYING or https://www.gnu.org/licenses/gpl-3.0.txt)
+# Copy this file to /usr/share/ansible/plugins/connection/pamssh.py

 from __future__ import (absolute_import, division, print_function)
 __metaclass__ = type

 DOCUMENTATION = '''
-    connection: ssh
+    connection: pamssh
     short_description: connect via ssh client binary
     description:
         - This connection plugin allows ansible to communicate to the target machines via normal ssh command
```

```
line.
@@ -210,6 +211,22 @@
           vars:
             - name: ansible_private_key_file
             - name: ansible_ssh_private_key_file
+      pamintegration:
+        description:
+          - Integration with PAM to get identity file. Name of the private key will be hostname.
+          - Name of Alias will be UNIXSSH-hostname-username. If the key retrival is not successful then use
user defined keys. If given will look for pampath  and run the python scripts.
+        ini:
+          - section: defaults
+            key: pamintegration
+        default: no
+      pampath:
+        description:
+          - Script location for A2A integration.
+          - Name of Alias will be UNIXSSH-hostname-username. If the key retrival is not successful then user
default key
+        ini:
+          - section: defaults
+            key: pampath
+        default: /usr/share/ansible/pamconnection.py

       control_path:
         description:
@@ -444,7 +461,7 @@
 class Connection(ConnectionBase):
     ''' ssh based connections '''

-    transport = 'ssh'
+    transport = 'pamssh'
     has_pipelining = True

     def __init__(self, *args, **kwargs):
@@ -547,6 +564,17 @@
         '''

         b_command = []
+        pamintegration = self.get_option('pamintegration')
+        pampath = self.get_option('pampath')
+        if( self.user == None ):
+          u='root'
+        else:
+          u=self.user
+        if pamintegration:
+          pamp = u"{0}   --host  {1}    --user   {2}".format(pampath,self.host, u)
+          os.system(pamp)
+        display.v(u'SSHCK: pamintegration  {0} pampath {1} command {2}'.format(pamintegration,pampath,pamp))
+

         #
         # First, the command to invoke
@@ -1158,7 +1186,7 @@

         super(Connection, self).exec_command(cmd, in_data=in_data, sudoable=sudoable)

-        display.vvv(u"ESTABLISH SSH CONNECTION FOR USER: {0}".format(self._play_context.remote_user),
host=self._play_context.remote_addr)
+        display.vvv(u"ESTABLISH A2ASSH CONNECTION FOR USER: {0}".format(self._play_context.remote_user),
host=self._play_context.remote_addr)

         if getattr(self._shell, "_IS_WINDOWS", False):
             # Become method 'runas' is done in the wrapper that is executed,
```

pamintegration plugin

pamintegration

```
#!/usr/bin/env python2.7
import commands
import os,time
import sys
import argparse

def getCredential(alias, cacheflag, optflag):
    cmd = "/opt/a2a/catech/cspmclient/bin/cspmclient" +" "+ alias+" "+cacheflag+" "+optflag
#   print cmd
    f=os.popen(cmd)
    j=args.folder + '/' + args.host
    o=open(j, 'w' )
    retVal= f.read()
    out =  retVal.split(' ',2)
    if ( out[0] == '400'):
      o.write(out[2])
      o.close()
    else:
### Setup a default credential or record failure.
      cmd = "/opt/a2a/catech/cspmclient/bin/cspmclient" +" "+ "UNIXSSH-centralserver-root" +" "+cacheflag+"
"+optflag
      f=os.popen(cmd)
      retVal= f.read()
      out =  retVal.split(' ',2)
      if ( out[0] == '400' ):
        o.write(out[2])
        o.close()
      else:
        return False
    os.chmod(j,0600)
    return True


if __name__ == "__main__":
    alias=""
    cacheflag=""
    optflag=""

    parser = argparse.ArgumentParser(description='Retrive Keys from PAM Vault.')
    parser.add_argument('--host', help='Name of the host',action='store',required='True')
    parser.add_argument('--user', help='User name',default='root',action='store')
    parser.add_argument('--folder', help='SSH Folder',action='store',default='/root/.sshpriv')
    parser.add_argument('--action', help='Command Action setup to setup the key and clean to clean up old keys',
nargs=1,default='setup',action='store')
    parser.add_argument('-bypasscache','-b' , help='By pass A2A agent cache',default=False,action='store_true')

    args = parser.parse_args()
    argc =  len(sys.argv)
    alias = 'UNIXSSH-' + args.host + '-' + args.user
    os.putenv('CSPM_CLIENT_HOME','/opt/a2a/catech')
    if args.bypasscache:
      cacheflag = '-b'

    if args.action == 'setup':
      retVal = getCredential(alias, cacheflag, optflag)
    if ( retVal == True ):
      exit(0)
    else:
      exit(1)
```

Configure Ansible to use A2A retrived keys

```
# vi /etc/ansible/ansible.cfg  ## Add below line
private_key_file = /root/.sshkeys/{{ inventory_hostname }}
==== How to use pamssh ====
To use this plugin on Ansible hosts  'ansibleclient' use below command

ansible-playbook  -c pamssh -i ansibleclient, a2a.yml

#
```