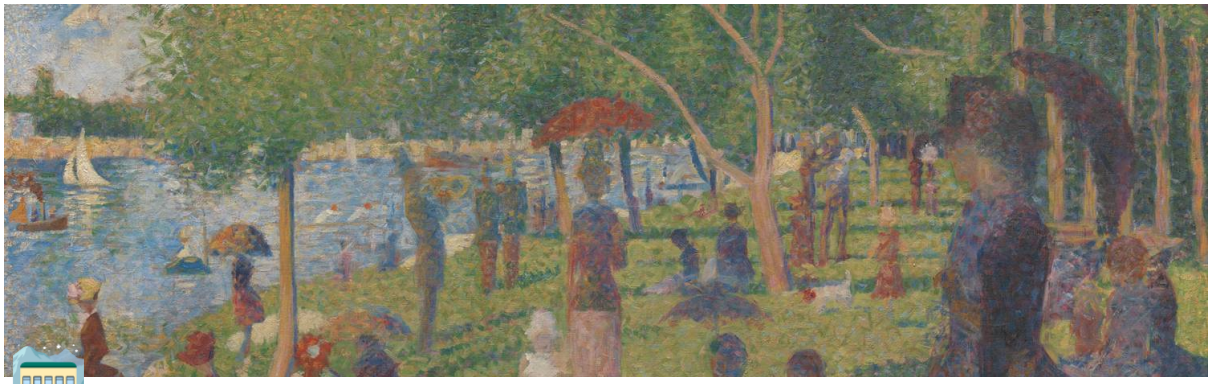


Radius Users Login

@July 13, 2022 @hyllydia

该测试用来测试Greatwall testbed 的Nssp11700 Radius Users Login

Step1,Create users in winserver（10.7.229.234） Siemax.com/Users， group：Radiususer

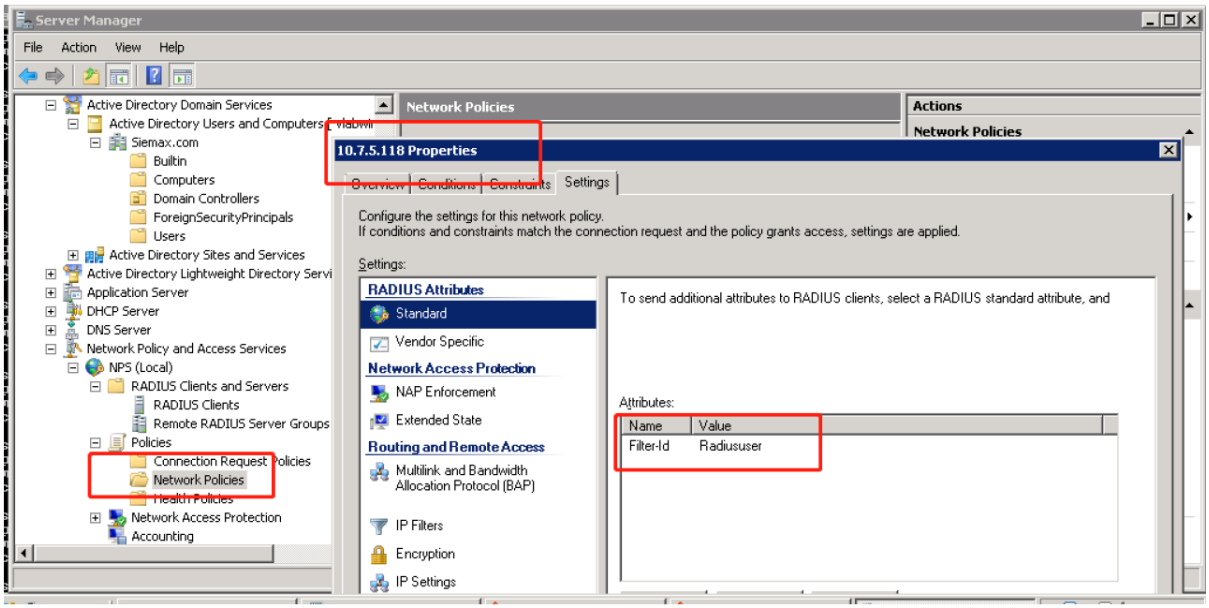
脚本生成CSV文件：

```
#coding:utf-8
"""
Author:Hou Yuling
Time:3/17/2022 3:26 PM
"""
#--coding:utf-8--
#思路：
#1、创建一个新的xlsx文件；
#2、workbook的名字
#3、A列,B列是用户名，user1, user2 ...依次加1
#4、C列是密码，D列域名，E列OU=Users, 这一列可以不写
#5、在盒子上测试，发现用户的CN="test1",CN="Users",DC="Siemax",DC="com"

import xlwt
import xlrd
import csv
import codecs

def Create_users():
    #新建一个文件；
    xls=xlwt.Workbook()
    sheet1=xls.add_sheet("sheet1")
    #添加字段
    sheet1.write(0,0,"First name")
    sheet1.write(0, 1, "Full name")
    sheet1.write(0, 2, "Password")
    sheet1.write(0,3,"Domain name")
    #循环添加内容
    for i in range(1,3001):
        sheet1.write(i,0,"user{}".format(i))
        sheet1.write(i,1,"user{}".format(i))
        sheet1.write(i,2,"hyllydia@2020")
        sheet1.write(i, 3, "Siemax.com")
    #保存文件
    xls.save("D:\create_radius_users.xls")
def xls_to_csv():
    workbook = xlrd.open_workbook('D:\create_radius_users.xls')
    table = workbook.sheet_by_name("sheet1")
    with codecs.open('D:\create_radius_users.csv', 'w',encoding="utf-8") as f:
        write = csv.writer(f)
        for row_num in range(table.nrows):
            write.writerow(table.row_values(row_num))

if __name__=="__main__":
      Create_users()
      xls_to_csv()
```

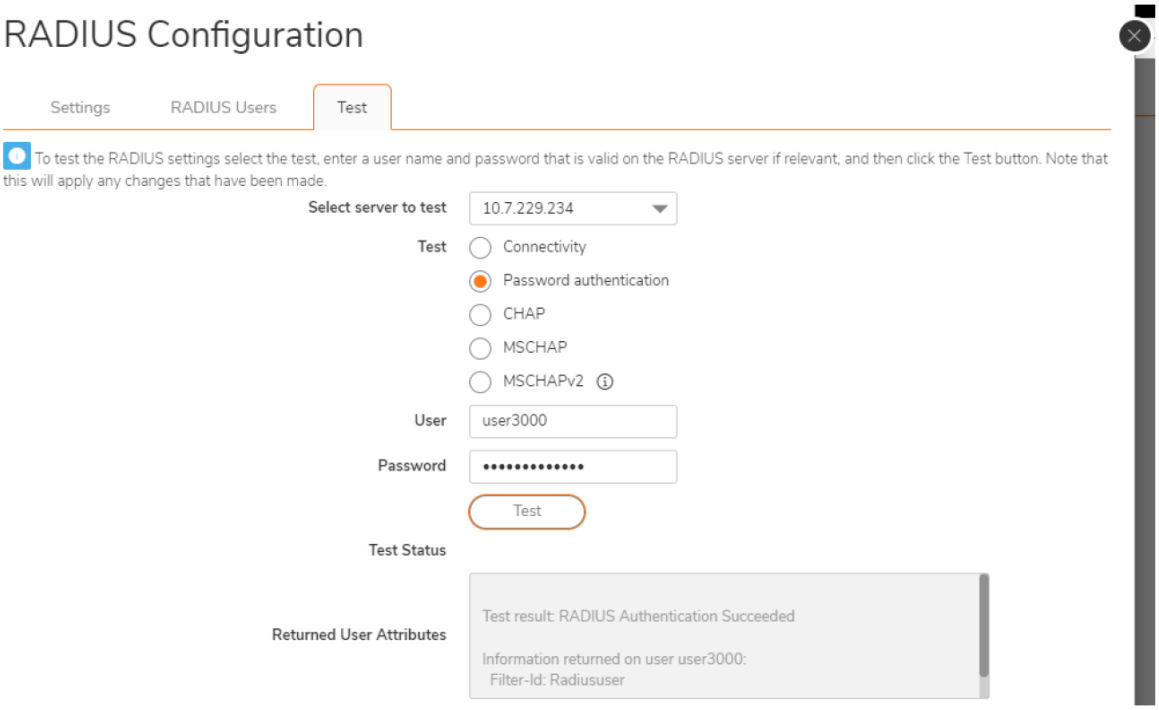
Step2，将CSV文件copy到Winserver上， 然后Winserver 上执行dos命令， 在Users Radiususer 这个group中生成用户：

```
for /f "skip=1 eol=; tokens=1,2,3,4 delims=, " %a in (c:\create_radius_users.csv) do dsadd user cn=%a,cn=Users,dc=Siemax,dc=com -displ
```

Step3, Radius Users Client:



Step4, 配置盒子， 参考LDAP Users Login 这篇笔记， 并在盒子上进行测试；



Step5, 在Nssp11700的X2 (VLAN 3594—s60 gi0/35)DMZ侧放置ubuntu虚拟机10.7.3.130 sonicwall/sonicwall

运行python scripts： python3 test.py 实现Radius users从DMZ侧登录盒子

test.py

```
#coding:utf-8
#1、subprocess第三方库用来运行命令，subprocess.Popen(cmd,shell=True)
#2、format()的用法
#3、xrange返回的是一个生成器，range返回的是一个列表；
#4.os.system()用来调用执行另外一个python文件，并且传递参数
import subprocess
```

```
import os
import datetime
import telnetlib

def deal_ab(a,b):
    if b>255:
        b=0
        a+=1
    return (a,b)
def add_virtual():
    a=1
    b=1
    for i in range(1,3001):
        print(i)
        a,b = deal_ab(a,b)
        cmd = "ifconfig ens192:{} 172.20.{}.{}/16".format(i,a,b)
        #cmd =("ifconfig ens192:%d 172.1.%d.%d/16" %(i,a,b))
        subprocess.Popen(cmd,shell=True)
        ip_login = "172.20.{}.{}".format(a,b)
        username = "user{}".format(i)
        os.system("python userlogin.py {} {}".format(ip_login,username))
        #x = ("%d %d" % (a,b))
        #print(x)
        b+=1
if __name__=="__main__":
    start_time = datetime.datetime.now()
    print("start:{}".format(start_time))
    add_virtual()
    end_time = datetime.datetime.now()
    print("end:{}".format(end_time))
```

userlogin.py

```
#coding:utf-8
#1、requests库
#2、sys.argv[]用来接收另一个python文件传递过来的参数；os.system("python 1.py {} {}".format(a,b))    a=sys.argv[1]   b=sys.argv[2]
#3、requests_toolbelt.adapters 用来设置虚拟IP，需要用不同的ip从同一个出口出去
import requests
from requests.auth import HTTPBasicAuth
from requests_toolbelt.adapters import source
import json
import sys
import urllib3
import time
urllib3.disable_warnings()

def user_login(ipstr,username):
        print(ipstr)
        print(username)
        s = requests.Session()
        new_source = source.SourceAddressAdapter(ipstr)
        s.mount('https://', new_source)
        url="https://172.20.254.254/api/sonicos/auth"
        payload={"override":'false',"snwl":'true'}
        data=json.dumps(payload)
        header={
                'Conection':'keep alive',
                'Accept':'application/json, text/plain, */*',
                'Content-Type': 'application/json;charset=UTF-8',
                'User-Agent':'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.116 Safari/5
                'Accept-Encoding':'gzip, deflate',
                'Accept-Language':'zh-CN,zh;q=0.8'
                }
        try:
            requests.packages.urllib3.disable_warnings()
            re = s.post(url, auth=HTTPBasicAuth(username, 'hyllydia@2020'), headers=header,data=data,verify=False)
            re.raise_for_status()
            print (re)
            if (re.status_code == 200):
                print("Login successful.")
        except requests.exceptions.RequestException as e:
                print ("- LoginERROR - Web service exception, msg  {}".format(e))
        #requests.packages.urllib3.disable_warnings()
        # r = s.post(url, auth=HTTPDigestAuth(username, 'password'), headers=header, data=data, verify=False)
        # print(r.status_code)
        time.sleep(2)

if __name__=="__main__":
        ipstr = sys.argv[1]
        username = sys.argv[2]
        user_login(ipstr,username)
```