from django.db.models import Count

from django.shortcuts import render, redirect

# Create your views here.

from Cyber\_Users.models import UserAdd\_Model

def admin\_login(request):

if request.method =="POST":

name = request.POST.get('name')

password = request.POST.get('password')

if name=='admin' and password == 'admin':

return redirect('user\_details')

return render(request, 'admins/admin\_login.html')

def achart\_page(request,chart\_type):

chart = UserAdd\_Model.objects.values('year').annotate(dcount=Count('organizationtype'))

return render(request,'admins/achart\_page.html',{'chart\_type':chart\_type,'objects':chart})

def admin\_analysis(request):

chart = UserAdd\_Model.objects.values('attackresult','method').annotate(dcount=Count('attackresult'))

return render(request,'admins/admin\_analysis.html',{'objects':chart})

def user\_details(request):

obj = UserAdd\_Model.objects.all()

return render(request,'admins/user\_details.html',{'object':obj})

import re

from django.contrib import messages

from django.contrib.auth import authenticate

from django.db.models import Q, Count

from django.shortcuts import render, redirect

# Create your views here.

from Cyber\_Users.forms import UserRegister\_Form

from Cyber\_Users.models import UserRegister\_Model, UserAdd\_Model

def user\_login(request):

if request.method == "POST":

name = request.POST.get('name')

password = request.POST.get('password')

try:

check = UserRegister\_Model.objects.get(name=name,password=password)

request.session['userid'] = check.id

return redirect('user\_adddata')

except:

pass

user = authenticate(name=name,password=password)

if user is not None:

if user.is\_active:

return redirect('user\_adddata')

else:

messages.error(request, 'username or password are not match')

return redirect('user\_login')

return render(request, 'users/user\_login.html')

def user\_register(request):

if request.method == "POST":

forms = UserRegister\_Form(request.POST)

if forms.is\_valid():

forms.save()

messages.success(request, 'You have been successfully registered')

return redirect('user\_login')

else:

forms = UserRegister\_Form()

return render(request,'users/user\_register.html',{'form':forms})

def user\_adddata(request):

userid = request.session["userid"]

obj = UserRegister\_Model.objects.get(id=userid)

attack1 = []

attack2, attack3, attack4, attack5, attack6, attack7, attack8, attack9 = [], [], [], [], [], [], [], []

splt = ''

Entity = ''

Year = 0

Records = ''

Organizationtype = ''

Method = ''

txt =''

Adddata = ''

ans = ''

Time = ''

if request.method == "POST":

Entity = request.POST.get("entity")

Year = request.POST.get("year")

Records = request.POST.get("records")

Organizationtype = request.POST.get("organizationtype")

Method = request.POST.get("method")

txt = request.POST.get("name")

Time = request.POST.get("time")

splt = (re.findall(r"[\w']+", str(txt)))

for f in splt:

if f in ('IPid', 'FDDI', 'x25', 'rangingdistance'):

attack1.append(f)

elif f in ('tcpchecksum', 'mtcp', 'controlflags', 'tcpoffset', 'tcpport'):

attack2.append(f)

elif f in ('ICMPID', 'udptraffic', 'udpunicorn', 'datagramid', 'NTP', 'RIP', 'TFTP'):

attack3.append(f)

elif f in ('GETID', 'POSTID', 'openBSD', 'appid', 'sessionid', 'transid', 'physicalid'):

attack4.append(f)

elif f in ('SYN', 'ACK', 'synpacket', 'sycookies'):

attack5.append(f)

elif f in ('serverattack', 'serverid', 'blockbankwidth'):

attack6.append(f)

elif f in ('monlist', 'getmonlist', 'NTPserver'):

attack7.append(f)

elif f in ('portid', 'FTPID', 'tryion', 'fragflag'):

attack8.append(f)

elif f in ('malwareid', 'gethttpid', 'httpid'):

attack9.append(f)

if len(attack1) > len(attack2) and len(attack1) > len(attack3) and len(attack1) > len(attack4) and len(

attack1) > len(attack5) and len(attack1) > len(attack6) and len(attack1) > len(attack7) and len(

attack1) > len(attack8) and len(attack1) > len(attack9):

ans = "Man-in-the-middle Attack"

elif len(attack2) > len(attack1) and len(attack2) > len(attack3) and len(attack2) > len(attack4) and len(

attack2) > len(attack5) and len(attack2) > len(attack6) and len(attack2) > len(attack7) and len(

attack2) > len(attack8) and len(attack2) > len(attack9):

ans = "Phishing and spear phishing attacks"

elif len(attack3) > len(attack2) and len(attack3) > len(attack1) and len(attack3) > len(attack4) and len(

attack1) > len(attack5) and len(attack1) > len(attack6) and len(attack1) > len(attack7) and len(

attack1) > len(attack8) and len(attack1) > len(attack9):

ans = "Drive-by attack"

elif len(attack4) > len(attack2) and len(attack4) > len(attack3) and len(attack4) > len(attack1) and len(

attack4) > len(attack5) and len(attack4) > len(attack6) and len(attack4) > len(attack7) and len(

attack4) > len(attack8) and len(attack4) > len(attack9):

ans = "Password attack"

elif len(attack5) > len(attack2) and len(attack5) > len(attack3) and len(attack5) > len(attack4) and len(

attack5) > len(attack1) and len(attack5) > len(attack6) and len(attack5) > len(attack7) and len(

attack5) > len(attack8) and len(attack5) > len(attack9):

ans = "SQL injection attack"

elif len(attack6) > len(attack2) and len(attack6) > len(attack3) and len(attack6) > len(attack4) and len(

attack6) > len(attack5) and len(attack6) > len(attack1) and len(attack6) > len(attack7) and len(

attack6) > len(attack8) and len(attack6) > len(attack9):

ans = "Cross-site scripting (XSS) attack"

elif len(attack7) > len(attack2) and len(attack7) > len(attack3) and len(attack7) > len(attack4) and len(

attack7) > len(attack5) and len(attack7) > len(attack6) and len(attack7) > len(attack1) and len(

attack7) > len(attack8) and len(attack7) > len(attack9):

ans = "Eavesdropping attack"

elif len(attack8) > len(attack2) and len(attack8) > len(attack3) and len(attack8) > len(attack4) and len(

attack8) > len(attack5) and len(attack8) > len(attack6) and len(attack8) > len(attack7) and len(

attack8) > len(attack1) and len(attack8) > len(attack9):

ans = "Birthday attack"

elif len(attack9) > len(attack2) and len(attack9) > len(attack3) and len(attack9) > len(attack4) and len(

attack9) > len(attack5) and len(attack9) > len(attack6) and len(attack9) > len(attack7) and len(

attack9) > len(attack8) and len(attack9) > len(attack1):

ans = "Teardrop attack"

else:

ans = "Unmalware"

UserAdd\_Model.objects.create(uregid=obj,entity=Entity,year=Year,records=Records,organizationtype=Organizationtype,method=Method,adddata=txt,attackresult=ans,time=Time)

return render(request,'users/user\_adddata.html')

def user\_page(request):

obj = UserAdd\_Model.objects.all()

return render(request,'users/user\_page.html',{'object':obj})

def malware(request):

obj = UserAdd\_Model.objects.filter(Q(attackresult='Man-in-the-middle (MitM) attack') | Q(attackresult='Phishing and spear phishing attacks') | Q(

attackresult='Drive-by attack') | Q(attackresult='Password attack') | Q(

attackresult='SQL injection attack') | Q(attackresult='Cross-site scripting (XSS) attack') | Q(attackresult='Eavesdropping attack') | Q(

attackresult='Birthday attack') | Q(attackresult='Teardrop attack'))

return render(request,'users/malware.html',{'object':obj})

def unmalware(request):

obj = UserAdd\_Model.objects.filter(attackresult='Unmalware')

return render(request,'users/unmalware.html',{'object':obj})

def breaches\_analysis(request):

chart = UserAdd\_Model.objects.values('attackresult','method').annotate(dcount=Count('attackresult'))

return render(request,'users/breaches\_analysis.html',{'objects':chart})

def chart\_page(request,chart\_type):

chart = UserAdd\_Model.objects.values('year').annotate(dcount=Count('organizationtype'))

return render(request,'users/chart\_page.html',{'chart\_type':chart\_type,'objects':chart})