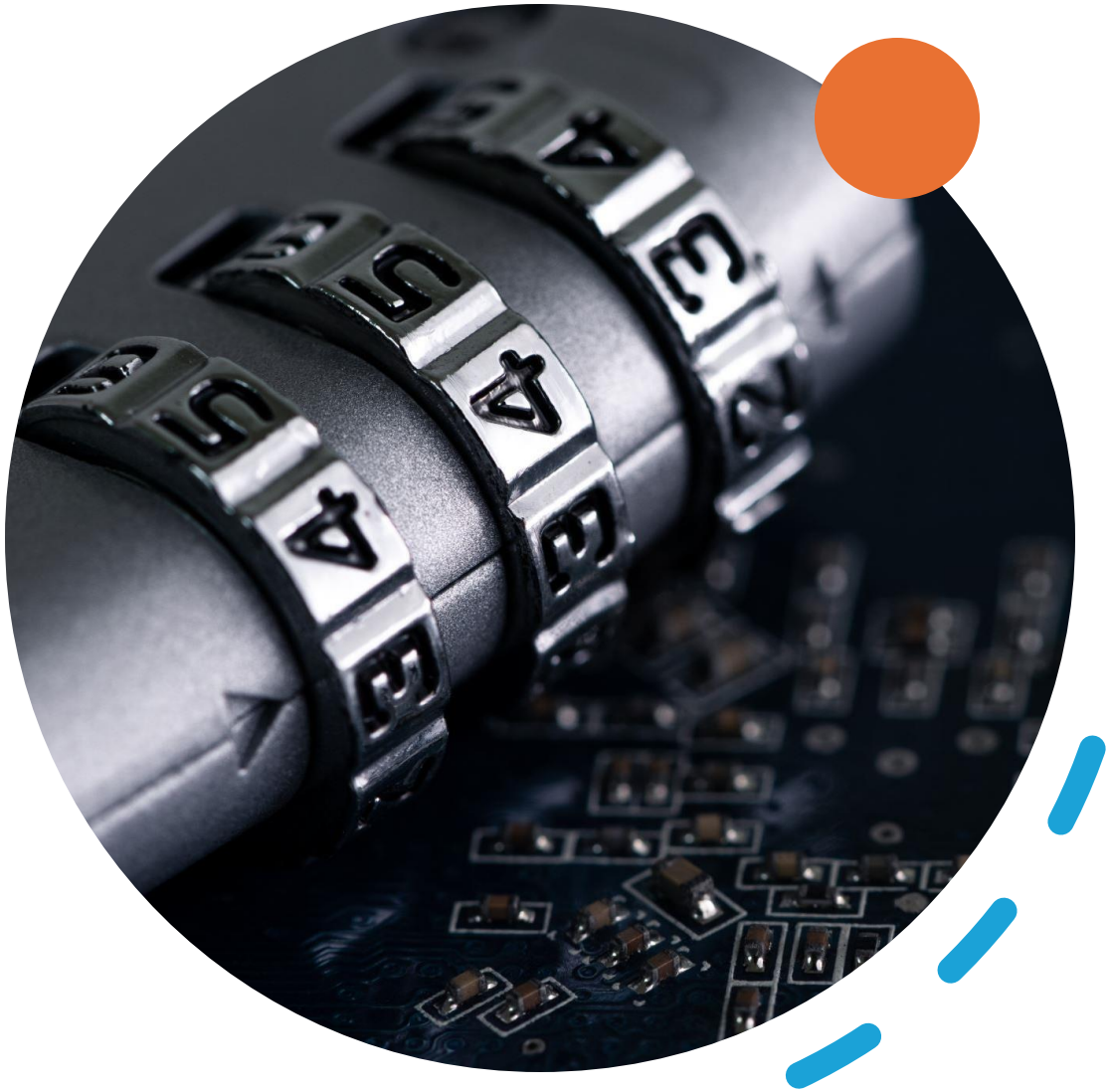
WEBCAM SPYWARE SECURITY

Code explanation presentation

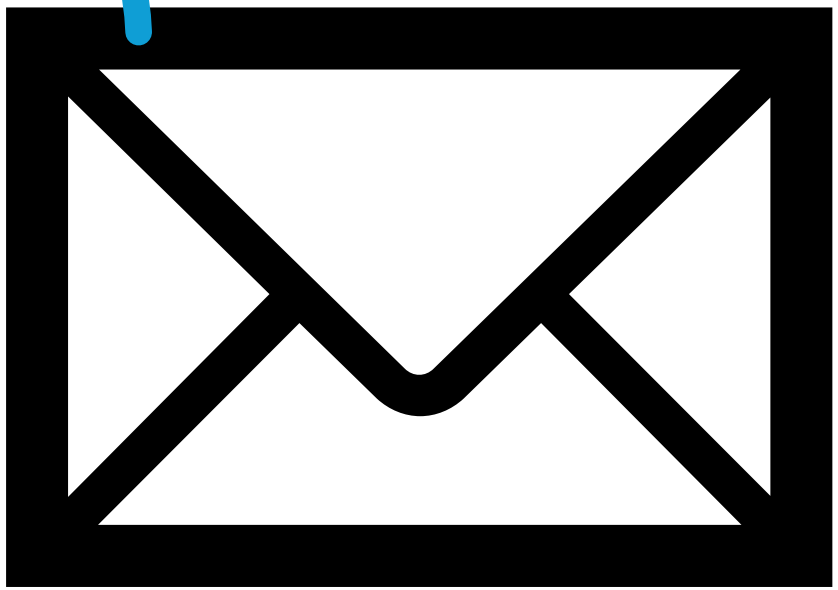
PACKAGES USED

- Tinker use for creating GUI
- Messagebox and simpledialog for showing messages ang dialog boxes
- Subprocess for running external scripts
- Json to convert json strings to python objects
- Os for interacting with operaating systems
- Smtplib and MIME TEXT for sending emails
- PIL for handling images
- Webbrowser to open urls
- CV2 is an opencv library used for video capturing



EMAIL SENDING FUNCTION

```
def send_email(subject, message):
    global email_settings
    try:
        msg = MIMEText(message)
        msg['Subject'] = subject
        msg['From'] = email_settings['EMAIL_ADDRESS']
        msg['To'] = email_settings['TO_EMAIL']


        with smtplib.SMTP('smtp.gmail.com', 587) as server:
            server.starttls()
            server.login(email_settings['EMAIL_ADDRESS'], email_settings['EMAIL_PASSWORD'])
            server.send_message(msg)
    except Exception as e:
        messagebox.showerror("Error", f"Failed to send email: {str(e)}")
```



- **send_email**: Composes and sends an email notification.
- Constructs the email message and uses Gmail's SMTP server to send it.
- Catches any exceptions and shows an error message if sending fails.

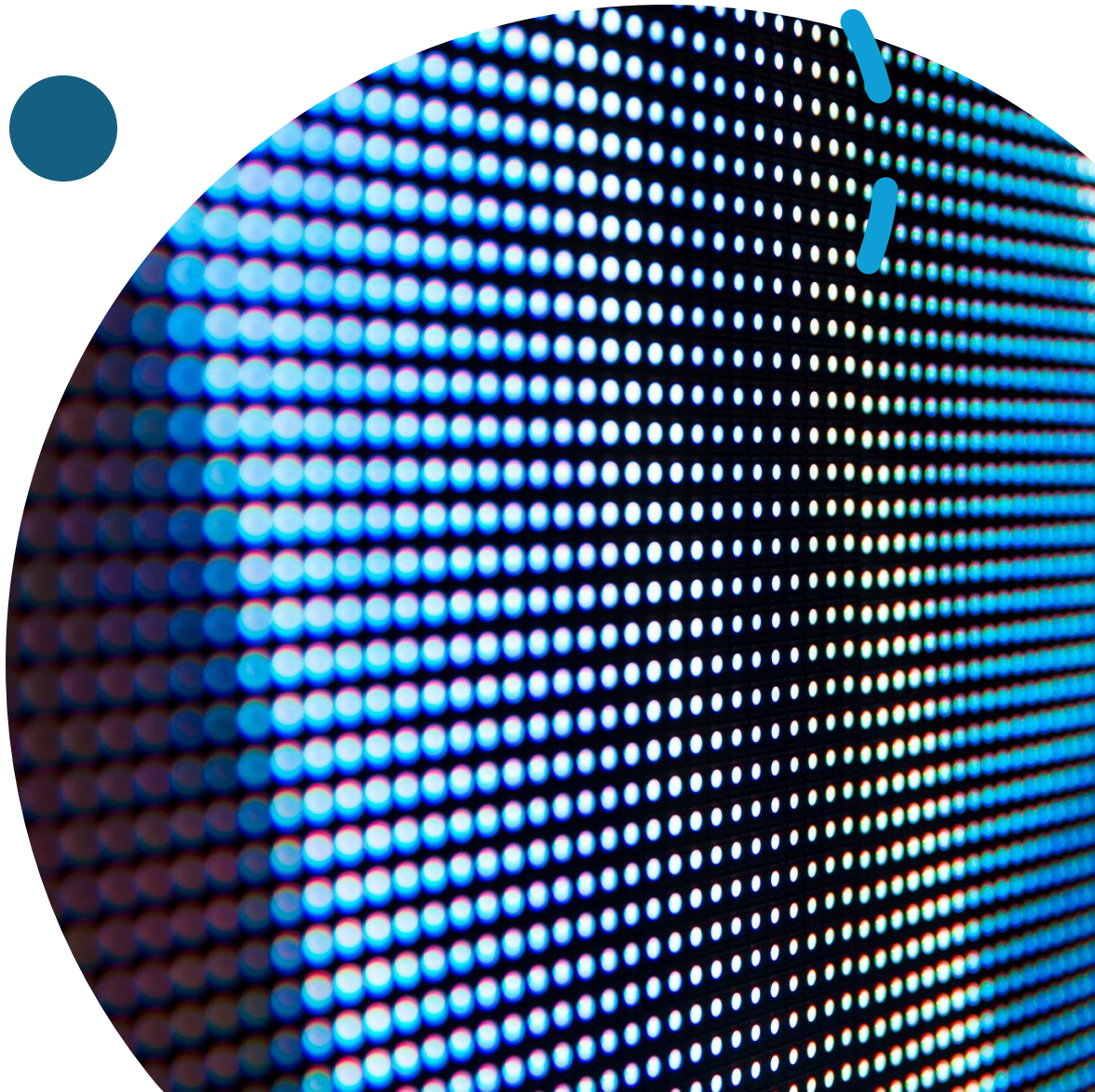
EMAIL ALERT

def send_email_alert(): subject = "Unauthorized Access Alert" message = "An incorrect password was entered." send_email(subject, message)

..... Send_email_alert sends a specific alert email as "unauthorized acess alert"

VIDEO CAPTURE FUNCTION

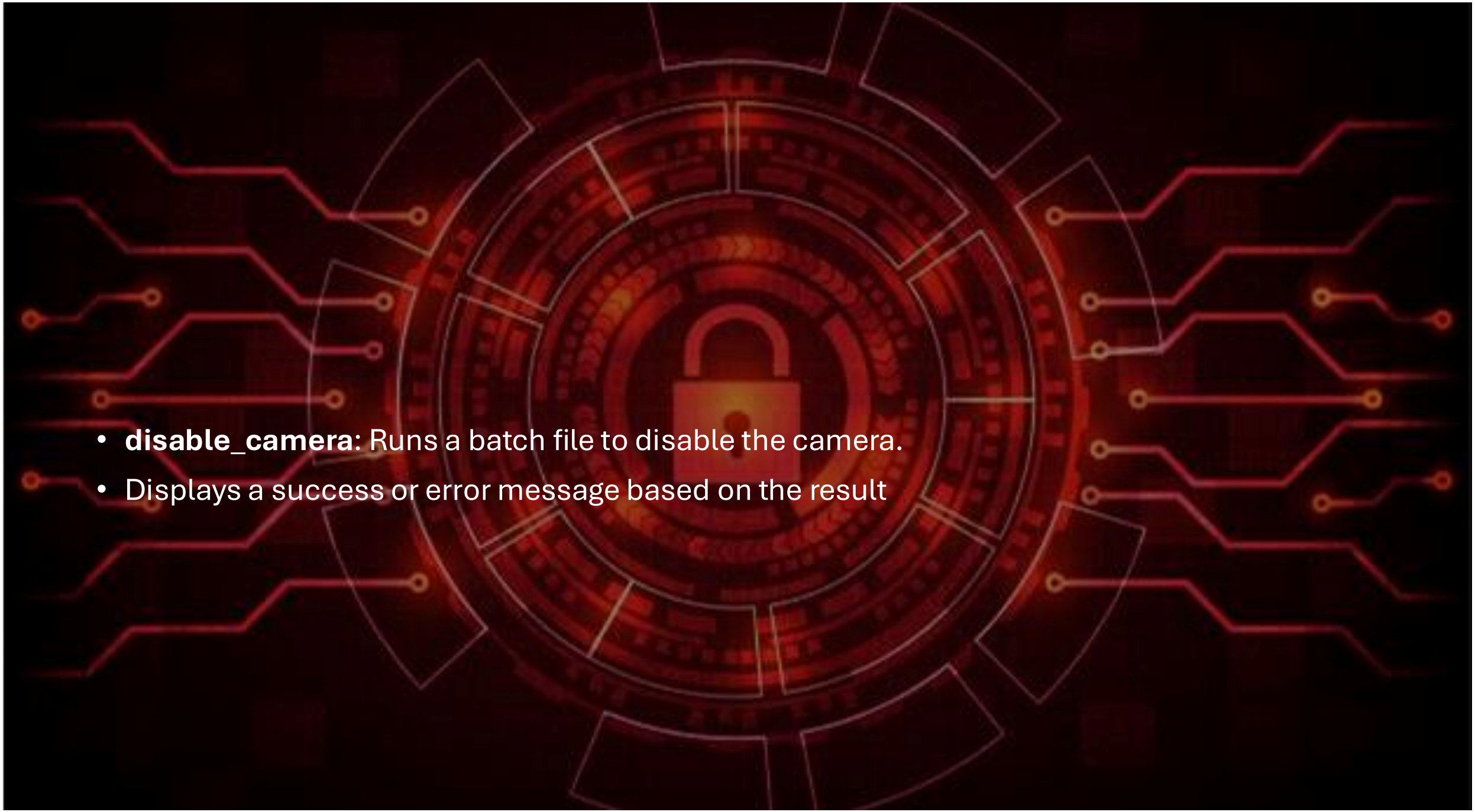
```
def capture_video():
    cap = cv2.VideoCapture(0)
    if not cap.isOpened():
        return

    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    out = cv2.VideoWriter('intrusion_video.avi', fourcc, 20.0, (640, 480))

    start_time = time.time()
    while int(time.time() - start_time) < 10:
        ret, frame = cap.read()
        if ret:
            out.write(frame)
        else:
            break

    cap.release()
    out.release()
    cv2.destroyAllWindows()
```



- Capture_video it captures 10 seconds video from the webcam
- It checks if the webcam is available then starts to capture 10 seconds of video
- Saves the video `intrusion_video.avi` and releases resources afterward.

CAMERA CONTROL FUNCTION

- def disable_camera():
- result = subprocess.run([r'disable_cam.bat'], shell=True)
- if result.returncode == 0:
- messagebox.showinfo("Success", "Camera disabled successfully.")
- else:
- messagebox.showerror("Error", "Failed to disable the camera.")



- **disable_camera**: Runs a batch file to disable the camera.
- Displays a success or error message based on the result