Layer 8+ Privacy

eSoc 488: Privacy Technology in Context

Due: 27 November 2018

Total Homework/Assignment Points: 100

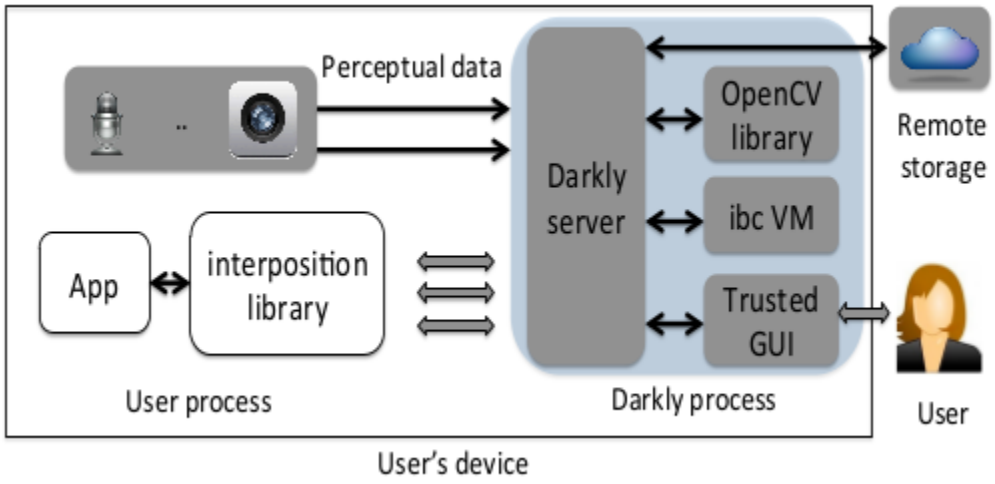
---

1 Layer 8+

Layer 8+ privacy problems arise when we consider networks as comprising not only computers, but humans as well. Here we think about layer 8+ issues in two particular contexts.

1.1 What are Layer 8+ Issues? How do they look concretely?

1. (14) Describe what a layer 8+ privacy issue is. How do layer 8+ issues arise in the PGP trust model? In standoff biometry?
2. (8) Describe how DARKLY allows an untrusted application to call an OpenCV function on a video frame from a camera. How does it address the layer 8+ privacy issues arising from passive physical access? In your description be sure to reference the following diagram [1]:



1.2 Hands on: Face Detection

Face detection and recognition from video surveillance is an important source of layer 8+ privacy problems. Face detection algorithms can be fooled, however—sometimes in unintuitive ways. Here we will set up facial recognition in python, and then experiment with fooling it. For this portion of the assignment you will find useful chapter five of *Learning OpenCV 3 Computer Vision with Python*, available at `http://proquestcombo.safaribooksonline.com.ezproxy4.library.arizona.edu/9781785283840/ch05_html`.

1. If you are not using the VM, install opencv. Otherwise, you may skip this step.
2. (8) Write a script called 'face_detect.py' to perform simple face detection in video. Use 'haarcascade_frontal_face_default.xml' as your classifier cascade file. Be sure to include copious, useful comments, and to write clear code.
3. (20) Read a little about how face detection using Haar Cascades works, and apply your understanding to try different things to reduce the confidence scores for the classification. Answer, in a file called 'anti-face-detection': Suppose this face detector were running on a surveillance camera that is out of your control. What actions could you take in the video to decrease the confidence of the classifier? What did you try? Can you find a method that would be workable for an everyday person walking in the street? *Hint: You will need to figure out how to get the confidence score from the detectMultiScale method.*
4. (20) Now suppose you can intercept the video stream. Implement a method to add noise to the video stream until your face is no longer detected. Try by experiment to find the minimum amount of noise to prevent detection, then save two images minimum_noise_detect.png and minimum_noise_nodetect.png showing images with noise that is close to this boundary on either side (i.e., one detecting a face, one not).
5. (10) Extra credit: Improve somehow the script given in the book to train a classifier to recognize a face in a video stream (not just detect a face), then use it to train a classifer to recognize your own face, and print your name above the box.

2 Diffie Hellman

Diffie-Hellman key exchange (DH) allows a secret to be securely communicated across a public channel; variants of DH are still widely used. Here we'll implement a simple version of DH.[1]

1. (20) Visit the class wiki page on setting up petlib.[2] Implement simple Diffie-Helmann key agreement protocol in a file called 'dh.py'. Write the key to a file to send it; read from the file to receive it. Use the key to encrypt the string 'squeamish ossifrage' with AES. Write the result to a file called 'aes_encrypted'.

References

[1] Suman Jana, Arvind Narayanan, and Vitaly Shmatikov. A Scanner Darkly: Protecting user privacy from perceptual applications. In *Security and Privacy (SP), 2013 IEEE Symposium on*, pages 349–363. IEEE, 2013.

[1] Since we didn't have an assignment for the communications privacy material, a question from that module is included here.

[2] `https://sidiprojects.us/mediawiki-1.23.2/index.php/Preparation_for_implementing_some_cryptographic_primitives_in_Python`