YOUR NAME

Assignment 2

eSoc 488: Information Privacy with Applications

Due: 23 November 2017

Total Homework/Assignment Points: 333

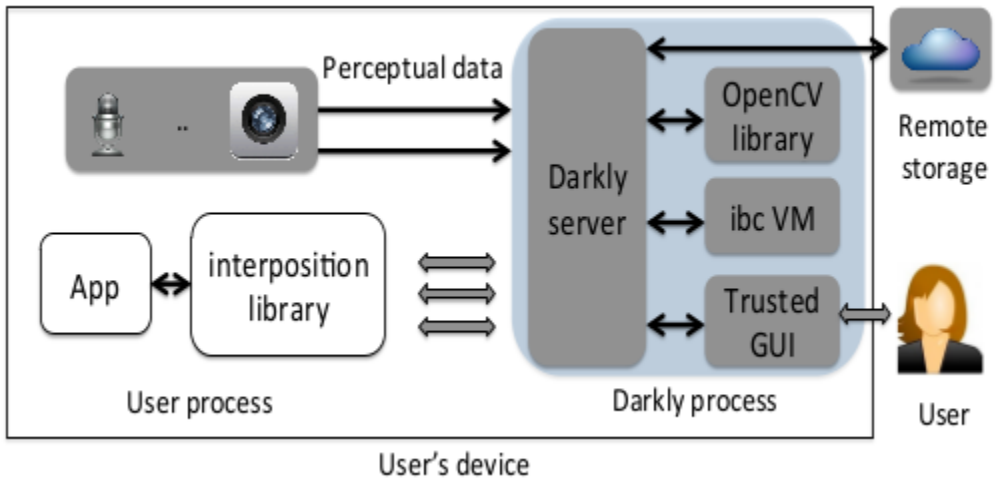
omissions List omissions here.
substitutions List substitutions here.
additions List additions here

---

1 Layer 8+

Layer 8+ problems arise in networks of humans and computers, even when the security properties of protocols at lower layers of the OSI model are preserved. Here we think about layer 8+ issues first abstractly, and then in a specific setting.

1. (30) Describe what a layer 8+ privacy issue is. How do layer 8+ issues arise in the PGP trust model? In standoff biometry?
2. (30) Describe how DARKLY allows an untrusted application to call an OpenCV function on a video frame from a camera. Base your description on the following diagram [1]:



2 Simple auditing

Even simple investigation of web traffic can be used to learn potentially-sensitive information. Here we set up a web server, and observe what information is visible to a few different perspectives.

1. (20) Set up a simple web server using the following steps:

(a) Set up a virtual private server (VPS) running Debian on Digitalocean.[1]

(b) Disable password authentication, and use an SSH key to access the server with a new account (not root).

(c) On your VPS, install ufw, and use it to ensure that your server drops all traffic that is not an SSH session from your current IP address, or a web request from 127.0.0.1 on port 8080.

(d) Install apache2. Edit the file /var/www/html/index.html to contain exactly the string 'DOWN WITH CYBERCRUD'.

2. (10) Look at your logs

   (a) Use tail -f to view the access logs on your server as they are changing. Visit your site. What information is visible about you? How could this be used to learn more?

3. (30) Use mitmproxy to record a flow for a site you're visiting (i.e., filter for just the traffic for the site itself, not third parties). Now do the same for a third party advertiser or analytics company on a website. Create a text dump of the third party flow as well, and search the text for something interesting.

3 Tor[2]

3.1 Onion Proxy

In this exercise we learn to interact with the onion proxy (OP) running on your client, and get information from it. You may wish to use the VM from last time for this part of the assignment, if you don't have another linux system to work on. First do the following:

- Open the control port of the Tor daemon using the Tor configuration file.[3]
- Set up a python script called 'tor_getinfo.py' that uses Stem to get information about your Tor connections.[4]

Now provide a detailed description of the steps you take to do the following:

1. (30) List the nodes used, with the IP-address and the fingerprint for each open Tor connection.
2. (10) Use the GeoIP Database[5] to find the locations of your Tor nodes.

3.2 Specifying parts of the circuit

Tor allows to define the entry and exit nodes by modifying the config file torrc. It is your task to use the following entry and exit nodes, which are identified by fingerprints.[6]

Guard nodes:

- B990B16CC9DD4709E256CFFFA22D3B3820DE53D7

[1] Or a similar service. These instructions don't assume anything about the VPS provider other than that it allows you to run a Debian instance. The choice of Debian is itself unimportant, other than to give us a specific distribution to work with.

[2] Thanks to Maximilian Golla at RUB for sharing the questions in sections 3.2 and 3.3 below, which I have modified for our purposes.

[3] The documentation can be found at `https://www.torproject.org/docs/tor-manual.html.en`.

[4] Stem documentation is at `https://stem.torproject.org/`.

[5] Found at `https://www.maxmind.com/en/geoip-demo`

[6] See footnote 3 above for the documentation.

- B70629C29A032979E5AD00CC51D2330845AAFD6A

Exit nodes:

- A53C46F5B157DD83366D45A8E99A244934A14C46
- 8AFDDF4B703C39152188F7A9A62B325B43EE8D84

Answer the following questions:

1. (30) List the nodes used, with the IP-address and the fingerprint for each open Tor connection and use the GeoIP Database[7] to find the location of your nodes.
2. (10) How does a strict selection of the Tor exit and entry nodes influence the anonymity of the connection?
3. (10) Is it possible to exclude nodes? When should this be done?

3.3 Tor Bridges

Bridges are Tor relays that are not listed in the main Tor directory. Since there is no complete public list of them, even if your ISP (Internet Service Provider) is filtering connections to all the known Tor relays, they probably would not be able to block all the bridges. If you suspect your access to the Tor network is being blocked, you may want to use the bridge feature of Tor. It is your task to establish a connection to the Tor network via a bridge.[8]

1. (30) Get a bridge that you can use to connect to the Tor network and give its IP, port and fingerprint. How did you get these pieces of information? What are other methods to get a bridge?
2. (10) In which situations is the application of bridges useful? Does the method to get a bridge that you chose before work in such situations?

3.4 Set up a Tor hidden service

Your next task is to set up a hidden service on the Tor network.

1. (50) Using your VPS, install tor, and set up a hidden service using the tor documention.[9] On your *local* computer, use the tor browser to visit your site (the address is found in $HOME/hidden_service/hostname). This site will need to remain available until you have received a grade for this assignment (not more than one week from the submission deadline). Turn in the onion address for your working hidden service in a file called 'hidden_service_address'.

Please answer the following questions about hidden services:

1. (11) How is the service hidden, i.e., why can users not reveal the location of the service (in the network) and still use it?
2. (11) Is there a way to still reveal information about the operators or the location of a hidden service? Monitor the connections to your hidden service (e.g., by watching the access log file).
3. (11) Can you distinguish between different users? Would this be different if your service was not a hidden service (and still accessed via Tor)?

[7] See the link in footnote 5.
[8] See `https://www.torproject.org/docs/bridges.html.en`
[9] See `https://www.torproject.org/docs/tor-hidden-service.html.en`

4 Extra credit

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. (30) Visit the class wiki page on setting up petlib.[10] 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.

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