Solution

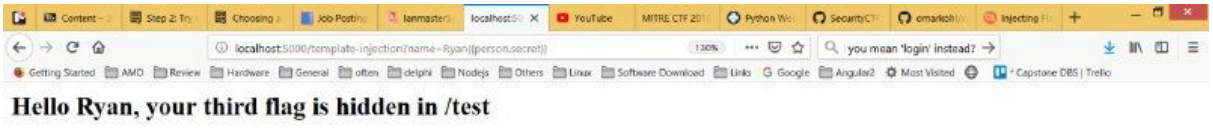
Part 1:
Use RSA decryption to find the plaintext needed to complete the link.

- Minimal brute force required
- Refer to Chinese Remainder Theorem for inspiration on the Math computation
- Refer to our python file for sample solution (crack.py)

Part 2:
Use either Inspect Element to find the hidden block leading to the next page OR use view page source to directly find the next link in the Javascript file script.js.

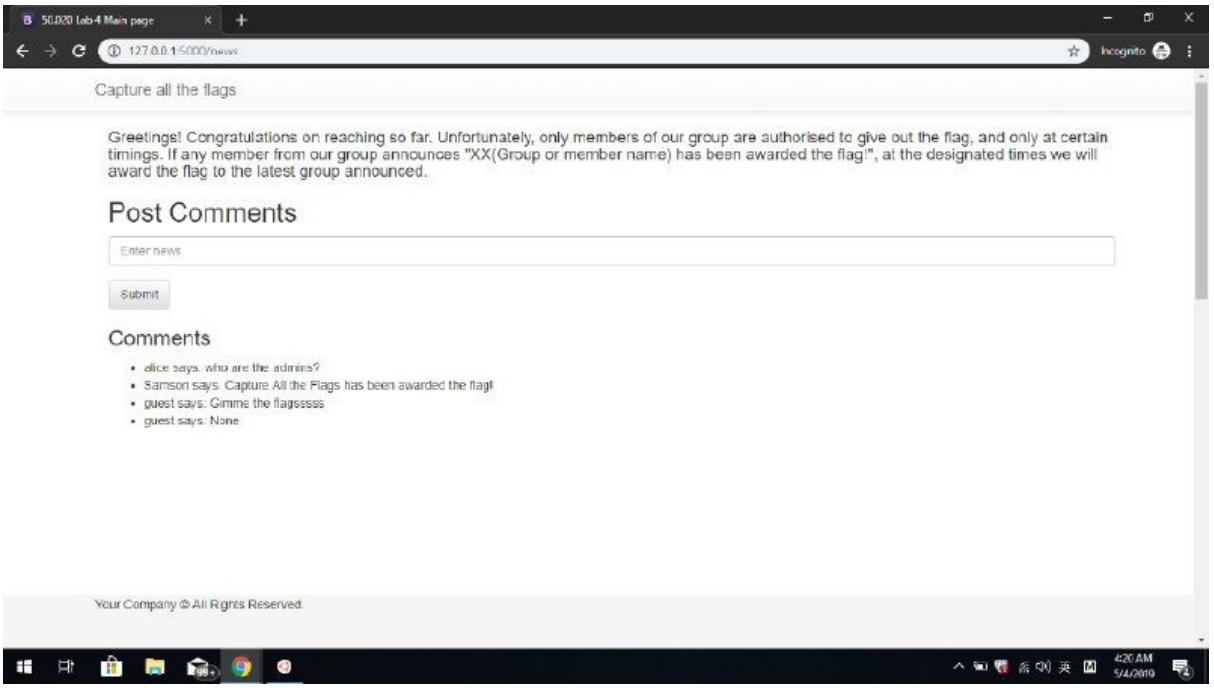
Part 3:
Use server-side-template-injection to find out the location of the next route.

- Refer to the link to find out that one is supposed to use SSTI
- Hint given to look for “person”’s “secret”
- Inject in link: {{person.secret}} to find out the instruction of the next location



Part 4:
Use HTML Injection Attack to post a comment as one of the group members.

- Inspect element to find out that each comment row is made from comment
- The format of each posted comment is “<USERNAME> says: <COMMENT>”
- Hence, the attacker may post the following comment to inject HTML code of a new line: “hello   Samson says: Bob the Builder has been awarded the flag!”
- Resulting HTML code posted will be: “
hello   Samson says: Bob the Builder has been awarded the flag!”



Part 5:

Use steganography to decrypt the image

- $ sudo apt update
- $ sudo apt-get install steghide
- $ cd <directory where you saved the image>
- $ steghide extract -sf "Mother Goose.jpg"
- Passphrase is *cyotCTF* (recovered through bruteforce attack of 7 alphabets)