Bandit

This worksheet introduces *many* new vocabulary terms. I am *encouraging* you to look these questions up online! We will go over these concepts in class; Bandit does not cover all of the material these questions introduce.

Model A | Bandit 11 to 15

1. What are some possible uses of base64 and rot13?

2. Which one is harder to crack, and why?

3. Write pseudocode for an algorithm that can bruteforce crack rot13. Your function should have the signature `break_rot13(ciphertext)` → `plaintext`

   Skip Bandit 12→13. It's a little tedious, and the password to `bandit13` is: `8ZjyCRiBWFYkneahHwxCv3wb2a1ORpYL`.

4. What are the permissions of `/etc/bandit_pass/bandit14`? Who is allowed to read that file?

5. What is an `ssh` key? How do you pass it as a parameter to `ssh`?

6. Explain the usage and syntax of the `nc` command. What does `nc` stand for?

7. Use `nc` for the following instructions:

   i. Send `"Hello world"` to `localhost` on port `8080`
   ii. Send `"secret"` to `box.secrets.org` on port `7777`
   iii. Send a file `password_file` to `no.access.net` on port `401`

8. How is a port different from an IP address?

Model B | Bandit 16 to 20

Using `openssl sclient` can be a bit confusing; it expects input on `stdin`. You can use that, or use piping to get around it. I also recommend using the `-quiet` flag!

1. What is SSL/TLS? What are sockets?
2. How does socket encryption help against exploitation?
3. How is `openssl s_client` similar to `nc`?
4. Explain the usage and syntax of `nmap`.
   i. How do you pass a range of ports into `nmap`?
   ii. What happens if you don't pass ports into `nmap`?
5. Use `nmap` for the following instructions:
   i. Scan ports `8000` to `9000` on server `horacemann.org`
   ii. Stealth scan ports `20` to `200` on server `black.box`
6. How do `nmap` stealth scans work?
7. What is TCP? What is the 3-step handshake?
8. Explain how to read the output of `diff`.
9. What is a `.bashrc` file?
10. How can you run a command via `ssh`?