


Branch: master ▾

Find file

Copy path

cs-35l / assignment8 / hw.txt




 prithvikannan added hw.txt with answers to questions
5475908 12 days ago

1 contributor

Raw

Blame

History

34 lines (28 sloc) 1.75 KB

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

1.

Since we are using SSH, which stands for secured shell, the channel is safe and even if other people could see bytes moving across, they would be encrypted using the symmetric key exchanged between both sides at the start of the session. The bytes would look like gibberish to the person watching because only the server and client have the key to decrypt messages.

If the malicious user had access to our keyboard after we have set up password-free authentication, that would not be an issue since even if they replicated the keystrokes, the server would not be able to verify they have the correct key (since we never typed the key in). If we had typed in the our login as username@host and password then they would be able to login as if they were me.

If the malicious user had access to our ssh files from the USB drive, then they would have the private key, which means that our security is breached and can use that to log in as if they were me.

2.

The gpg2 --verify command only checks that the file with the detached signature matches the key, but never checks if I was the one who wrote the key. For example, when I ran --gen-key I could have put someone else's email address and pretended to be them. A malicious actor could potentially create their own detached signature after manipulating the file and the end user would not know. All the verify command does is check the file and file signature.

We can fix this problem by adding another layer to verify that I am who I say I am when creating the signature. For example, I can encrypt a message using my private key, and they can look up the public key that is associated with my name, and then attempt to decrypt the message using that public key. If they are able to, that proves that I am who I say I am.