Vulnerability Report

Cryptography and Network Security I

*CSCI-4971-01*

**Taha Mehdi** **Pratik Patel** **Chris Renus**

**Power Outage Attack**

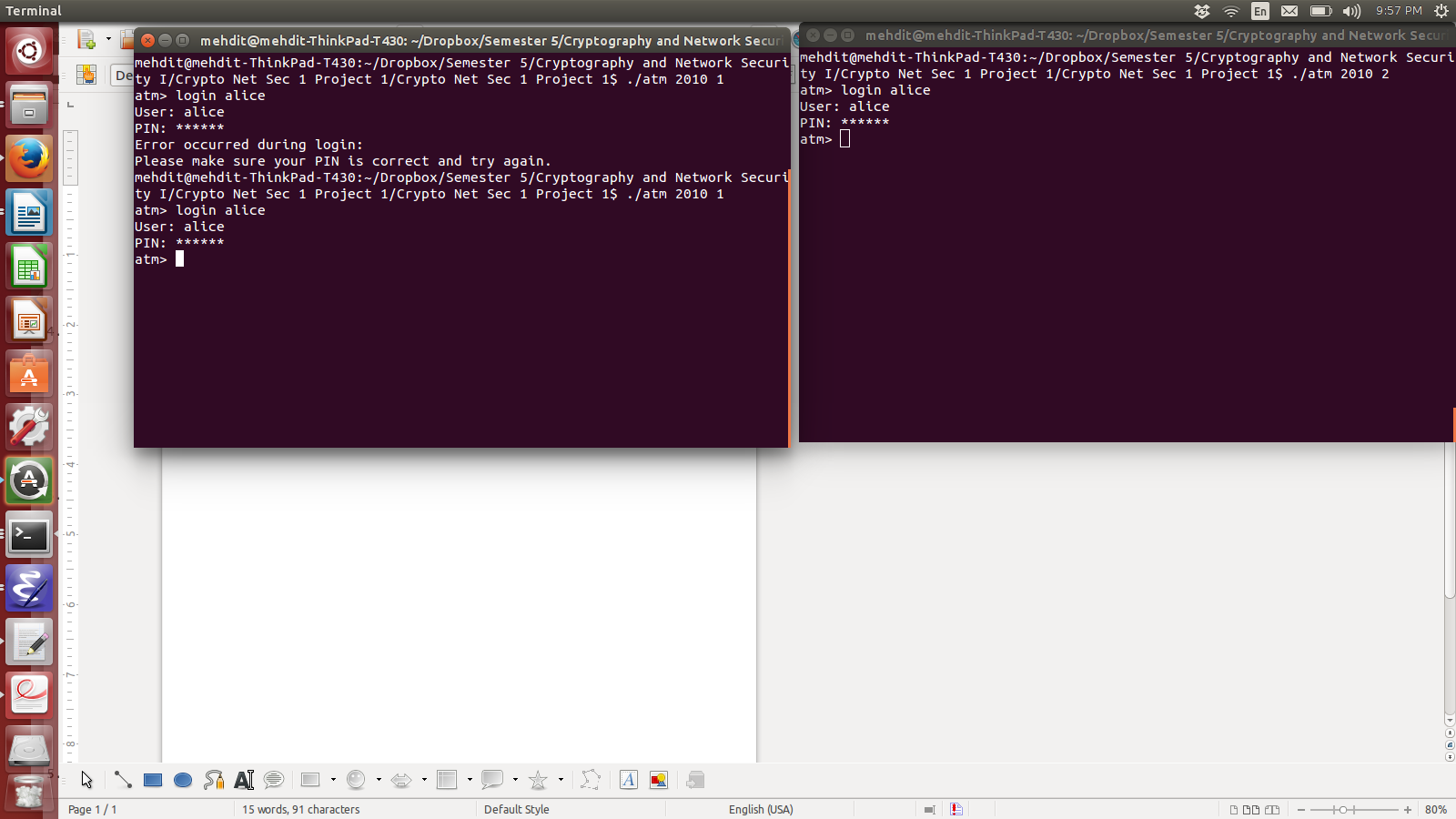
The other team stores each account in a vector that resides in a “Bank” class. Due to these accounts residing in memory, when the bank is closed all accounts are wiped. Therefore, we can in turn create a power outage attack against the bank (EMP attack) and force the computer running the bank server to close. Now all accounts will be deleted and since there is no backup of accounts, the bank cannot know what the accounts balances were.

**Concurrency Issues**

The accounts do not have a lock to prevent multiple ATMs to write to balance at the same time. If a user A is transferring money to user B, and at the same time user C is transferring money to user B there can be race conditions. If user B initially has $1000, both A and C can read that value and transfer $500. Therefore, instead of user B ending up with $2000, they will only have $1500. Another possible condition is user A login from 2 ATMs and has a balance of $100. It is possible to withdraw $100 from each ATM and actually withdraw $200 in total.

**Multiple ATM Sessions**

The other team allows the same user to login at multiple ATMs. This is invalid since there should only be one ATM card per user and so allows for copies of cards to be created.



**Denial of Service (DoS) Attack**

In this protocol there can only be 50 ATMs running at the same time. Since there is no stopping a user from logging in from multiple ATMs, we can create a DoS attack by logging in to every ATM. Since there is no timeout feature to logout a user, once there are 50 ATMs occupied, no other user can use the bank.

**Bank Commands**

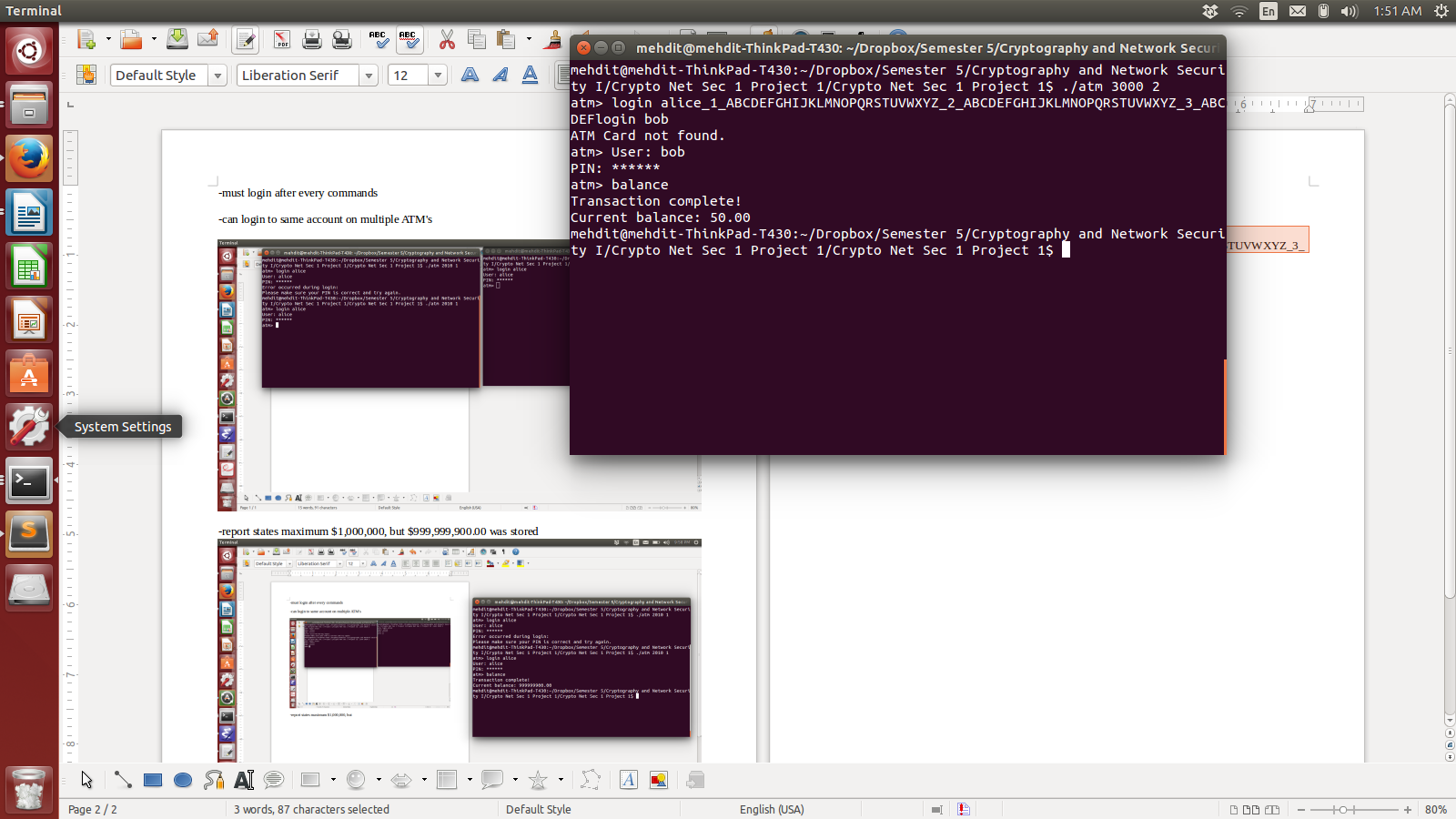
Through the bank terminal by using the balance command, it tells whether the account is valid or not. Through this, the team is able to gain information on all possible users within the bank.

**Bank Usage Issues**

The team has found many features that will detriment usage for the end user. For example, every time a command is entered at an ATM, the user is then logged out. This is poor user experience and does not emulate a real-world ATM. There exists a deposit, transfer, and withdraw limit for each individual account. These limits include 3 transfer attempts, $1,000 total withdraw limit, and $1,000,000,000 total deposit limit. This is an issue because you are able to deposit more than you can withdraw therefore allowing the bank to keep your money.

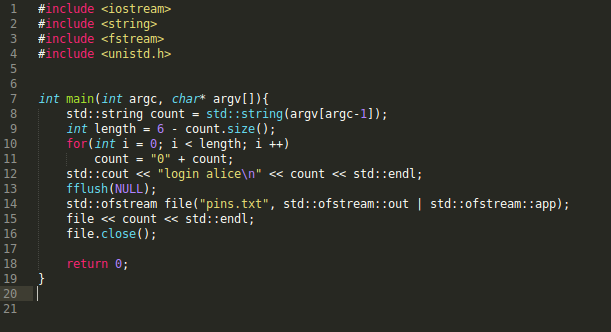
**Multiple Command Input**

The input buffer for the command is of length 80. Because of the looping structure of how the code is read, a command such as login alice\_1\_ABCDEFGHIJKLMNOPQRSTUVWXYZ\_2\_ABCDEFGHIJKLMNOPQRSTUVWXYZ\_3\_ABCDEFlogin bob (i.e. a valid input command, followed by a string such that the length of the command plus the length of the string plus 1 for the space is equal to 78) yields for the possibility to execute simultaneous commands.



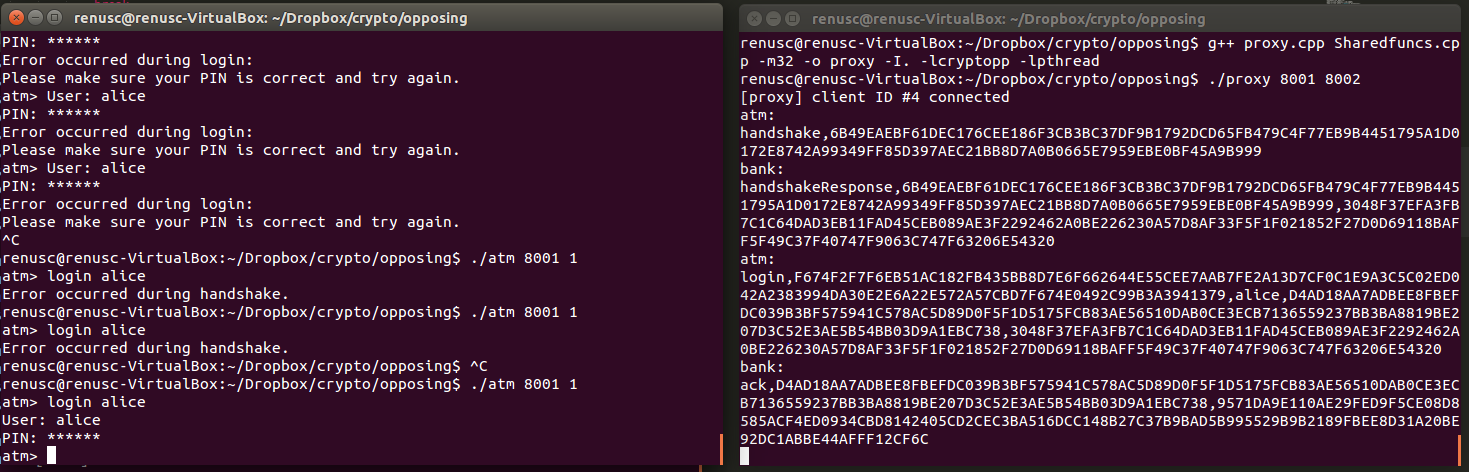
**Infinite Login Attempts**

The other team has not taken the precautions to limit the number of times a user can attempt to login to an account. Therefore we are able to crack these pins by trying various number of pins (000000 - 999999). The team wrote code (seen below) to crack the pins; the approximate time to attempt 5000 pins was 10 minutes, so all possible pins would be recovered in around 33 hours using a laptop.



**AES Keys**

The AES keys are stored in a file called keys. Upon creation of the bank, all keys are generated and stored within this file and stored onto the ATM. This means that through purchasing an ATM, the keys can be extracted. The image below on the right shows the decrypted packet. All packets can be seen in plaintext along with the nonce.



**Nonces**

The nonces do not serve much purpose. They are not a hash of the plaintext and so there is no checking of tampering. The nonces are just transferred to the other side without being a hash of anything and can be copied and used to act as the other side. The nonces are not dependent upon anything during the initial handshake and so there is no authentication or validation so an adversary could impersonate the bank, create a nonce, and the ATM would have no idea. Therefore, with the AES cracked, this would allow for a user on the ATM to login to any account, with any pin, and withdraw all of the money from the ATM without ever contacting the bank.