

I. INTRODUCTION

The goal of this project is to harden password-based authentication by including information obtained from a second factor (2-Factor Authentication or 2FA). To explore this, the Linux login command will be implemented. The following are the learning objectives of this project:

- Understand how password-based authentication is implemented.
- Understand the benefits of multiple factors for stronger authentication.
- **Task1:** Augment a password-based login command code to include an additional factor obtained from a second source. This section will include the results from test cases that verify the 2FA implementation.
- **Task2:** Analyze security benefits (or lack of them) of the 2FA implementation.

II. Task1: Implementing 2FA

In this task, a 2-Factor Authentication (2FA) application is developed using the provided token generator (TG) executable, which serves as a second factor. The 2FA method uses the tokens generated by TG to harden the login mechanism used in Linux. In Linux, typically, a unique second factor source/device is associated with a user (i.e., your phone for Duo 2FA used by Georgia Tech) but in this project, the same TG will serve as a second factor source for all accounts you create. For this to be possible, a user must be registered with TG along with a PIN. Thus, each user has two accounts.

a. Test Cases

This section will display results from running nine test cases (test case #1 doesn't count).

1. Include screenshot showing contents of /etc/passwd, /etc/shadow and home folder.

2. Create a user account of Ann with the following credentials:

Username: Ann

Password: Ann@123456789

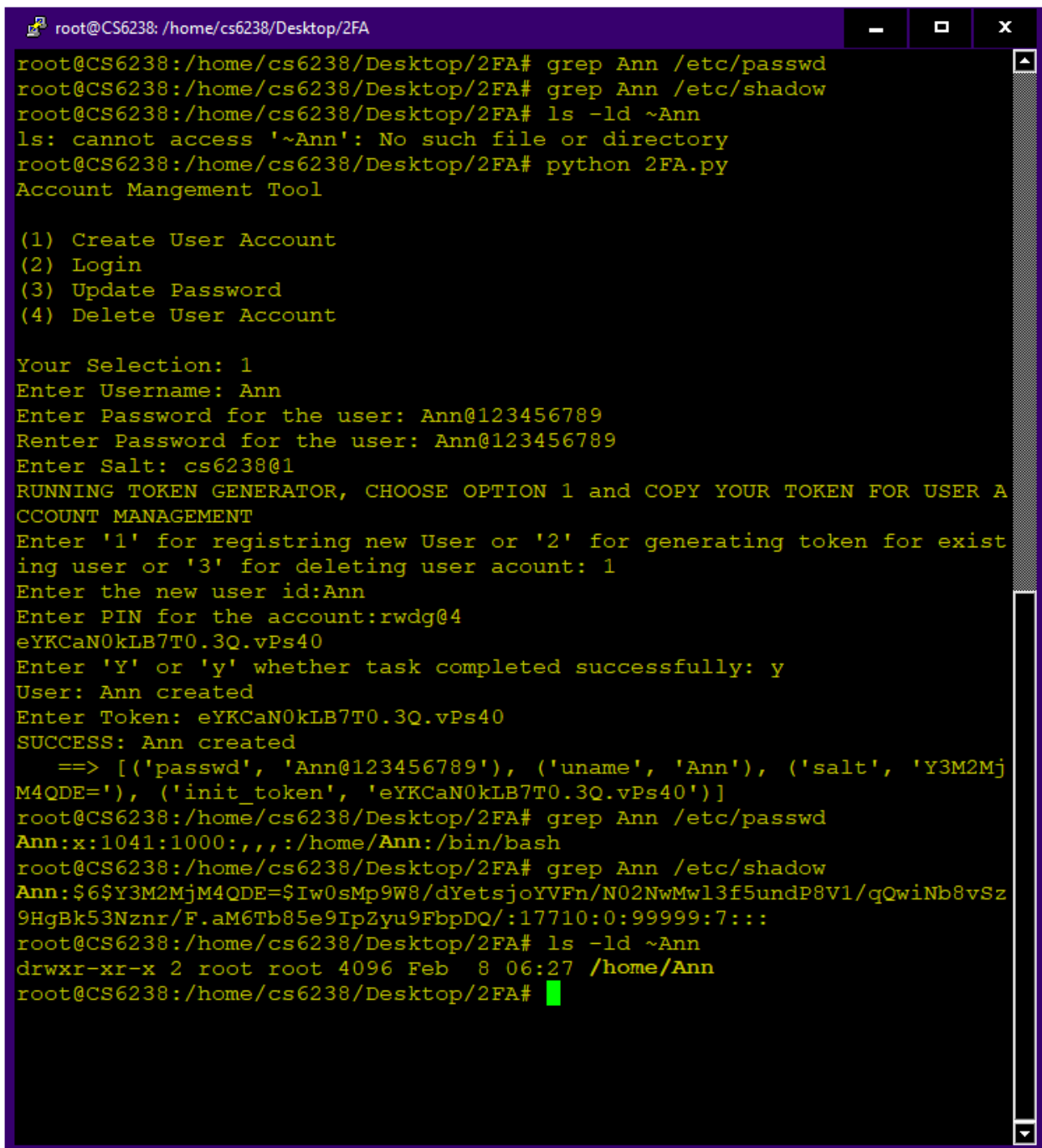
Salt: cs6238@1

Userid: Ann

Pin: rwdg@4

SUCCESS: Ann created

```
==> [('passwd', 'Ann@123456789'), ('uname', 'Ann'), ('salt', 'Y3M2MjM4QDE='), ('init_token', 'eYKCaN0kLB7T0.3Q.vPs40')]
```



```
root@CS6238: /home/cs6238/Desktop/2FA
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/passwd
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/shadow
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ann
ls: cannot access '~Ann': No such file or directory
root@CS6238:/home/cs6238/Desktop/2FA# python 2FA.py
Account Mangement Tool

(1) Create User Account
(2) Login
(3) Update Password
(4) Delete User Account

Your Selection: 1
Enter Username: Ann
Enter Password for the user: Ann@123456789
Renter Password for the user: Ann@123456789
Enter Salt: cs6238@1
RUNNING TOKEN GENERATOR, CHOOSE OPTION 1 and COPY YOUR TOKEN FOR USER A
CCOUNT MANAGEMENT
Enter '1' for registring new User or '2' for generating token for exist
ing user or '3' for deleting user account: 1
Enter the new user id:Ann
Enter PIN for the account:rwdg@4
eYKCaN0kLB7T0.3Q.vPs40
Enter 'Y' or 'y' whether task completed successfully: y
User: Ann created
Enter Token: eYKCaN0kLB7T0.3Q.vPs40
SUCCESS: Ann created
==> [('passwd', 'Ann@123456789'), ('uname', 'Ann'), ('salt', 'Y3M2Mj
M4QDE='), ('init_token', 'eYKCaN0kLB7T0.3Q.vPs40')]
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/passwd
Ann:x:1041:1000:,,,:/home/Ann:/bin/bash
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/shadow
Ann:$6$Y3M2MjM4QDE=$Iw0sMp9W8/dYetsjoYVFN/N02NwMw13f5undP8V1/qQwiNb8vSz
9HgBk53Nznr/F.aM6Tb85e9IpZyu9FbpDQ/:17710:0:99999:7:::
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ann
drwxr-xr-x 2 root root 4096 Feb  8 06:27 /home/Ann
root@CS6238:/home/cs6238/Desktop/2FA#
```

3. Create a user account of Ben with following credentials:

Username: Ben

Password: Ben@123456

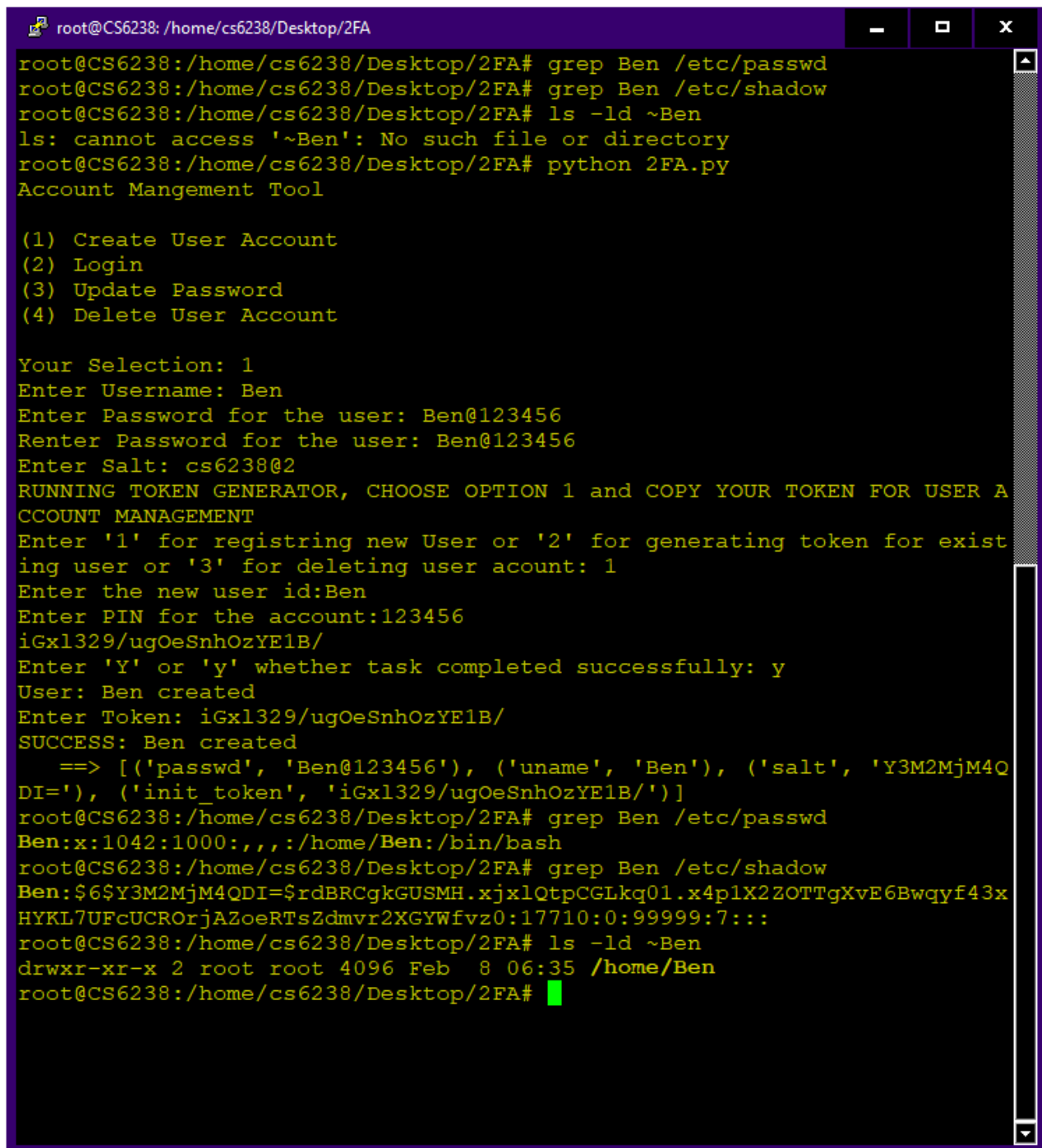
Salt: cs6238@2

Userid: Ben

Pin: 123456

SUCCESS: Ben created

```
==> [('passwd', 'Ben@123456'), ('uname', 'Ben'), ('salt', 'Y3M2MjM4QDI='), ('init_token', 'iGxl329/ugOeSnhOzYE1B/')]
```



```
root@CS6238: /home/cs6238/Desktop/2FA
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/passwd
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/shadow
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ben
ls: cannot access '~Ben': No such file or directory
root@CS6238:/home/cs6238/Desktop/2FA# python 2FA.py
Account Mangement Tool

(1) Create User Account
(2) Login
(3) Update Password
(4) Delete User Account

Your Selection: 1
Enter Username: Ben
Enter Password for the user: Ben@123456
Renter Password for the user: Ben@123456
Enter Salt: cs6238@2
RUNNING TOKEN GENERATOR, CHOOSE OPTION 1 and COPY YOUR TOKEN FOR USER A
CCOUNT MANAGEMENT
Enter '1' for registring new User or '2' for generating token for exist
ing user or '3' for deleting user account: 1
Enter the new user id:Ben
Enter PIN for the account:123456
iGxl329/ugOeSnhOzYE1B/
Enter 'Y' or 'y' whether task completed successfully: y
User: Ben created
Enter Token: iGxl329/ugOeSnhOzYE1B/
SUCCESS: Ben created
==> [('passwd', 'Ben@123456'), ('uname', 'Ben'), ('salt', 'Y3M2MjM4Q
DI='), ('init_token', 'iGxl329/ugOeSnhOzYE1B/')]
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/passwd
Ben:x:1042:1000:,,,:/home/Ben:/bin/bash
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/shadow
Ben:$6$Y3M2MjM4QDI=$rdBRCgkGUSMH.xjxlQtpCGLkq01.x4p1X2ZOTTgXvE6Bwqyf43x
HYKL7UFcUCROrjAZoERTsZdmvr2XGYWfvz0:17710:0:99999:7:::
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ben
drwxr-xr-x 2 root root 4096 Feb  8 06:35 /home/Ben
root@CS6238:/home/cs6238/Desktop/2FA#
```

4. Try and create another account for Ann with same TG credentials:

Username: Ann

Password: Ann@987654321

Salt: cs6238@3

Userid: Ann

Pin: rwdg@4

Your Selection: 1

Enter Username: Ann

FAILURE: user Ann already exists. Try deleting it first.

```
root@CS6238: /home/cs6238/Desktop/2FA
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/passwd
Ann:x:1041:1000:::/home/Ann:/bin/bash
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/shadow
Ann:$6$Y3M2MjM4QDE=$Iw0sMp9W8/dYetsjoYVFn/N02NwMw13f5undP8V1/qQwiNb8vSz9HgBk53Nz
nr/F.aM6Tb85e9IpZyu9FbpDQ/:17710:0:99999:7:::
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ann
drwxr-xr-x 2 root root 4096 Feb  8 08:48 /home/Ann
root@CS6238:/home/cs6238/Desktop/2FA# python 2FA.py
Account Mangement Tool

(1) Create User Account
(2) Login
(3) Update Password
(4) Delete User Account

Your Selection: 1
Enter Username: Ann
FAILURE: user Ann already exists. Try deleting it first.
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/passwd
Ann:x:1041:1000:::/home/Ann:/bin/bash
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/shadow
Ann:$6$Y3M2MjM4QDE=$Iw0sMp9W8/dYetsjoYVFn/N02NwMw13f5undP8V1/qQwiNb8vSz9HgBk53Nz
nr/F.aM6Tb85e9IpZyu9FbpDQ/:17710:0:99999:7:::
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ann
drwxr-xr-x 2 root root 4096 Feb  8 08:48 /home/Ann
root@CS6238:/home/cs6238/Desktop/2FA#
```

5. Try to login into Ben account with following credential:

Username: Ben

Password: Ben@123456

Userid: Ben

Pin: 123456

Enter the user id:Ben

Enter PIN for the Ben :123456

('iGx1329/ugOeSnhOzYE1B/', 'XRcyVF5J5jYa/YHCYoKdl.')

Enter 'Y' or 'y' whether task completed successfully: y

Enter Next Token: XRcyVF5J5jYa/YHCYoKdl.

SUCCESS: Ben logged in

```
root@CS6238: /home/cs6238/Desktop/2FA
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/passwd
Ben:x:1043:1000:,,,:/home/Ben:/bin/bash
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/shadow
Ben:$6$Y3M2MjM4QDI=$rdBRCgkGUSMH.xjxlOtpCGLkq01.x4p1X2ZOTtgXvE6Bwqyf43xHYKL7UFcU
CROrjAZoeRTsZdmvr2XGYWfvz0:17710:0:99999:7:::
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ben
drwxr-xr-x 2 root root 4096 Feb  8 10:09 /home/Ben
root@CS6238:/home/cs6238/Desktop/2FA# python 2FA.py
Account Mangement Tool

(1) Create User Account
(2) Login
(3) Update Password
(4) Delete User Account

Your Selection: 2
Enter Username: Ben
Enter Password for the Ben: Ben@123456
Enter Token: iGx1329/ugOeSnhOzYE1B/
RUNNING TOKEN GENERATOR, CHOOSE OPTION 2 and COPY YOUR NEW TOKEN FOR USER ACCOUNT MANAGEMENT
Enter '1' for registering new User or '2' for generating token for existing user or '3' for deleting user account: 2
Enter the user id:Ben
Enter PIN for the Ben :123456
('iGx1329/ugOeSnhOzYE1B/', 'XRcyVF5J5jYa/YHCYoKdl.')
```

Enter 'Y' or 'y' whether task completed successfully: y

Enter Next Token: XRcyVF5J5jYa/YHCYoKdl.

SUCCESS: Ben logged in

```
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/passwd
Ben:x:1043:1000:,,,:/home/Ben:/bin/bash
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/shadow
Ben:$6$Y3M2MjM4QDI=$G2lZK6Pc7.mCTuF5Zp46IR97zbJpMaNGaD7V3QRqhfhuhPk/N30DHet3pX/Td
6OJEQI1GMN3Yi32eE3DzPk.FR1:17710:0:99999:7:::
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ben
drwxr-xr-x 2 root root 4096 Feb  8 10:09 /home/Ben
root@CS6238:/home/cs6238/Desktop/2FA#
```

6. Try to login into Ann account with following credential:

Username: Ann

Password: Ben@123456

Userid: Ann

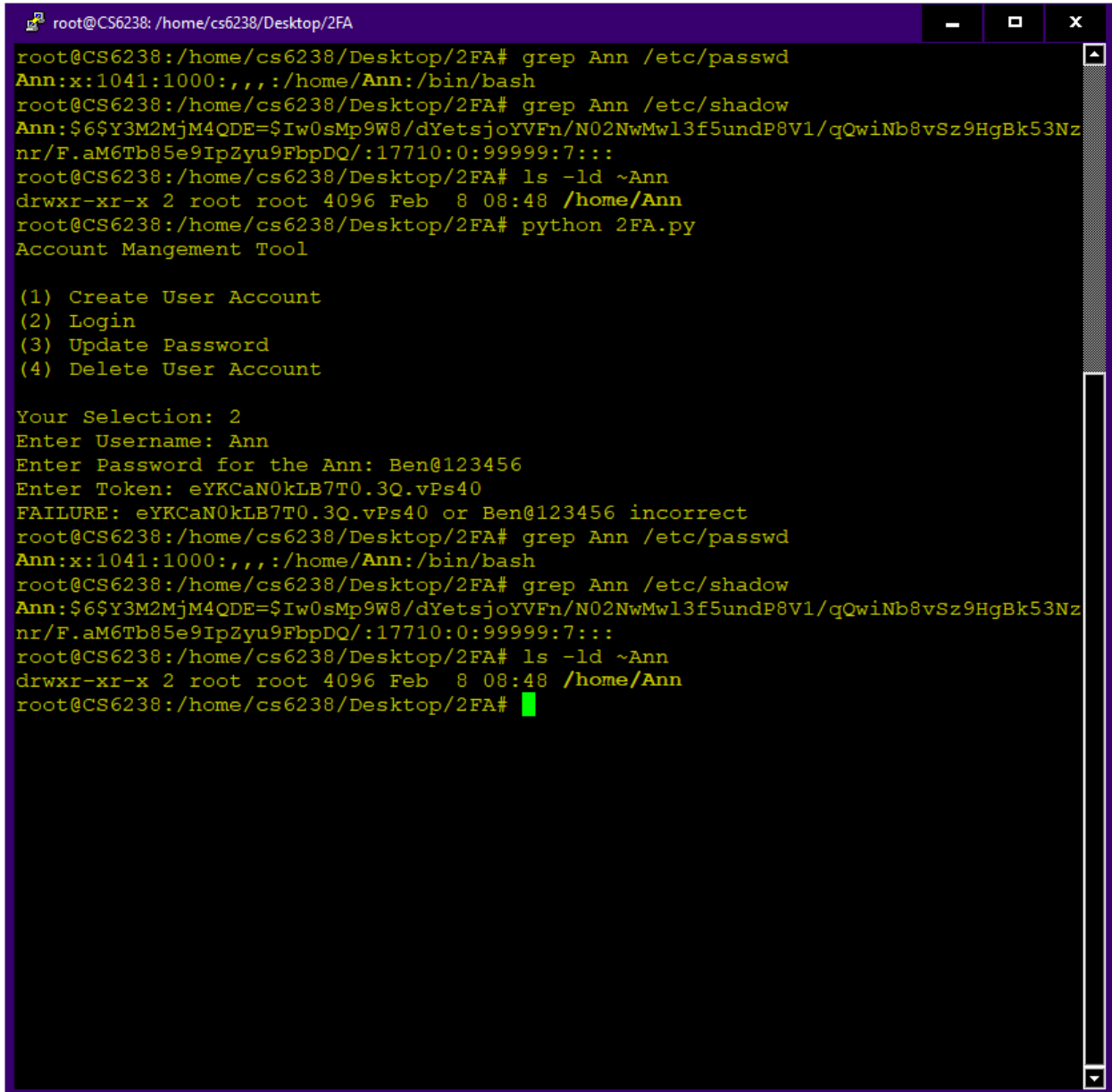
Pin: rwdg@4

Enter Username: Ann

Enter Password for the Ann: Ben@123456

Enter Token: eYKCaN0kLB7T0.3Q.vPs40

FAILURE: eYKCaN0kLB7T0.3Q.vPs40 or Ben@123456 incorrect



```
root@CS6238: /home/cs6238/Desktop/2FA
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/passwd
Ann:x:1041:1000:,,,:/home/Ann:/bin/bash
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/shadow
Ann:$6$Y3M2MjM4QDE=$Iw0sMp9W8/dYetsjoYVFfn/N02NwMw13f5undP8V1/qQwiNb8vSz9HgBk53Nz
nr/F.aM6Tb85e9IpZyu9FbpDQ/:17710:0:99999:7:::
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ann
drwxr-xr-x 2 root root 4096 Feb  8 08:48 /home/Ann
root@CS6238:/home/cs6238/Desktop/2FA# python 2FA.py
Account Mangement Tool

(1) Create User Account
(2) Login
(3) Update Password
(4) Delete User Account

Your Selection: 2
Enter Username: Ann
Enter Password for the Ann: Ben@123456
Enter Token: eYKCaN0kLB7T0.3Q.vPs40
FAILURE: eYKCaN0kLB7T0.3Q.vPs40 or Ben@123456 incorrect
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/passwd
Ann:x:1041:1000:,,,:/home/Ann:/bin/bash
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/shadow
Ann:$6$Y3M2MjM4QDE=$Iw0sMp9W8/dYetsjoYVFfn/N02NwMw13f5undP8V1/qQwiNb8vSz9HgBk53Nz
nr/F.aM6Tb85e9IpZyu9FbpDQ/:17710:0:99999:7:::
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ann
drwxr-xr-x 2 root root 4096 Feb  8 08:48 /home/Ann
root@CS6238:/home/cs6238/Desktop/2FA#
```

7. Try to update Ann Account:

Username: Ann

Password: Ann@123456789

New Password: georgiatech

New Salt: gatech01

Userid: Ann

Pin: rwdg@4

Enter the user id:Ann

Enter PIN for the Ann :rwdg@4

('eYKCaN0kLB7T0.3Q.vPs40', 'Y09JEiExPDjq.ndOhswKE/')

Enter 'Y' or 'y' whether task completed successfully: y

Enter Next Token: Y09JEiExPDjq.ndOhswKE/

SUCCESS: Password updated for Ann

```
root@CS6238: /home/cs6238/Desktop/2FA
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/passwd
Ann:x:1041:1000:,,,:/home/Ann:/bin/bash
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/shadow
Ann:$6$Y3M2MjM4QDE=$Iw0sMp9W8/dYetsjoYVFN/N02NwMw13f5undP8V1/qQwiNb8vSz9HgBk53Nz
nr/F.aM6Tb85e9IpZyu9FbpDQ/:17710:0:99999:7:::
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ann
drwxr-xr-x 2 root root 4096 Feb  8 08:48 /home/Ann
root@CS6238:/home/cs6238/Desktop/2FA# python 2FA.py
Account Mangement Tool

(1) Create User Account
(2) Login
(3) Update Password
(4) Delete User Account

Your Selection: 3
Enter Username: Ann
Enter Password for the Ann: Ann@123456789
Enter Token: eYKCaN0kLB7T0.3Q.vPs40
Enter a new Password for Ann: georgiatech
Renter Password for the user: georgiatech
Enter a new salt for Ann: gatech01
RUNNING TOKEN GENERATOR, CHOOSE OPTION 2 and COPY YOUR NEW TOKEN FOR USER ACCOUNT MANAGEMENT
Enter '1' for registering new User or '2' for generating token for existing user or '3' for deleting user account: 2
Enter the user id:Ann
Enter PIN for the Ann :rwdg@4
('eYKCaN0kLB7T0.3Q.vPs40', 'Y09JEiExPDjq.ndOhswKE/')
Enter 'Y' or 'y' whether task completed successfully: y
Enter Next Token: Y09JEiExPDjq.ndOhswKE/
SUCCESS: Password updated for Ann
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/passwd
Ann:x:1041:1000:,,,:/home/Ann:/bin/bash
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/shadow
Ann:$6$Y3M2MjM4QDE=$GuRwv2UR3GPKcWooVKfOHY0owwRfw4fUGDULfZDa0wxuFkvrYwtgzK1MuqAO
7sLK/Kgyct9vFnts9Q0NQe2Gu0:17710:0:99999:7:::
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ann
drwxr-xr-x 2 root root 4096 Feb  8 08:48 /home/Ann
root@CS6238:/home/cs6238/Desktop/2FA#
```


8. Try to update Ben Account: *When you login to the token generator, you will get a pair of tokens ('CT', 'NT') in that order. Only for this test case, input the NT obtained from the token generator as the CT requested by the 2FA scheme. For the NT requested by 2FA also input the NT given by TG.*
- Username: Ben
Password: Ben@123456
New Password: techgeorgia
New Salt: gatech02
Userid: Ben
Pin: 123456
- I implemented running the Token Generator (TG) into the 2FA.py application. I also confirm user credentials for login or update before running the TG (to obtain the next token). Thus, for this test case, I will first invoke the TG to get the next token.*

Enter Username: Ben
Enter Password for the Ben: Ben@123456
Enter Token: hEzIy//8/B3dgclJhVzAQ.
FAILURE: hEzIy//8/B3dgclJhVzAQ. or Ben@123456 incorrect

```
root@CS6238: /home/cs6238/Desktop/2FA
root@CS6238:/home/cs6238/Desktop/2FA# ./token_generator
Enter '1' for registering new User or '2' for generating token for existing user
or '3' for deleting user account: 2
Enter the user id: Ben
Enter PIN for the Ben :123456
('XRcyVF5J5jYa/YHCYoKdl.', 'hEzIy//8/B3dgclJhVzAQ.')
Enter 'Y' or 'y' whether task completed successfully: y
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/passwd
Ben:x:1043:1000:::/home/Ben:/bin/bash
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/shadow
Ben:$6$Y3M2MjM4QDI=$G2lZK6Pc7.mCTuF5Zp46IR97zbJpMaNGaD7V3QRqhfuHpk/N30DHet3pX/Td
6OJEQI1GMN3Yi32eE3DzPk.FR1:17710:0:99999:7:::
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ben
drwxr-xr-x 2 root root 4096 Feb  8 10:09 /home/Ben
root@CS6238:/home/cs6238/Desktop/2FA# python 2FA.py
Account Mangement Tool

(1) Create User Account
(2) Login
(3) Update Password
(4) Delete User Account

Your Selection: 3
Enter Username: Ben
Enter Password for the Ben: Ben@123456
Enter Token: hEzIy//8/B3dgclJhVzAQ.
FAILURE: hEzIy//8/B3dgclJhVzAQ. or Ben@123456 incorrect
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/passwd
Ben:x:1043:1000:::/home/Ben:/bin/bash
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/shadow
Ben:$6$Y3M2MjM4QDI=$G2lZK6Pc7.mCTuF5Zp46IR97zbJpMaNGaD7V3QRqhfuHpk/N30DHet3pX/Td
6OJEQI1GMN3Yi32eE3DzPk.FR1:17710:0:99999:7:::
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ben
drwxr-xr-x 2 root root 4096 Feb  8 10:09 /home/Ben
root@CS6238:/home/cs6238/Desktop/2FA#
```


9. Delete Ann account:

Username: Ann

Password: georgiatech

Userid: Ann

Pin: rwdg@4

Enter the user id:Ann

Enter PIN for the Ann :rwdg@4

Y09JEiExPDjq.ndOhswKE/

Enter 'Y' or 'y' whether task completed successfully: y

SUCCESS: Ann deleted

```
root@CS6238: /home/cs6238/Desktop/2FA
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/passwd
Ann:x:1041:1000:,,,:/home/Ann:/bin/bash
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/shadow
Ann:$6$Y3M2MjM4QDE=$GuRWv2UR3GPKcWooVKfoHY0owwRfw4fUGDULfZDa0wxuFkvrYwtgzKlMuqAO
7sLK/Kgyct9vFnts9Q0NQe2Gu0:17710:0:99999:7:::
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ann
drwxr-xr-x 2 root root 4096 Feb  8 08:48 /home/Ann
root@CS6238:/home/cs6238/Desktop/2FA# python 2FA.py
Account Mangement Tool

(1) Create User Account
(2) Login
(3) Update Password
(4) Delete User Account

Your Selection: 4
Enter Username: Ann
Enter Password for the Ann: georgiatech
Enter Token: Y09JEiExPDjq.ndOhswKE/
RUNNING TOKEN GENERATOR, CHOOSE OPTION 3
Enter '1' for registring new User or '2' for generating token for existing user
or '3' for deleting user account: 3
Enter the user id:Ann
Enter PIN for the Ann :rwdg@4
Y09JEiExPDjq.ndOhswKE/
Enter 'Y' or 'y' whether task completed successfully: y
SUCCESS: Ann deleted
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/passwd
root@CS6238:/home/cs6238/Desktop/2FA# grep Ann /etc/shadow
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ann
ls: cannot access '~Ann': No such file or directory
root@CS6238:/home/cs6238/Desktop/2FA#
```

10. Delete Ben account with:

Username: Ben

Password: Ben@123456

Userid: Ben

Pin:123456

Enter the user id:Ben

Enter PIN for the Ben :123456

hEzIy//8/B3dgclJhVzAQ.

Enter 'Y' or 'y' whether task completed successfully: y

SUCCESS: Ben deleted

This test case spawned an addition test case (see Appendix, Test Case A1). Recall that Test Case 8 attempted to update Ben's account. However, this failed due to an invalid token (next token: hEzIy//8/B3dgclJhVzAQ.). Since Ben's account was not updated, deleting the account requires using the current token (XRcyVF5J5jYa/YHCYoKdl.). Note that Ben's account was updated by the TG, but not the 2FA.py application.

```
root@CS6238: /home/cs6238/Desktop/2FA
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/passwd
Ben:x:1043:1000:,,,:/home/Ben:/bin/bash
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/shadow
Ben:$6$Y3M2MjM4QDI=$G2lZK6Pc7.mCTuF5Zp46IR97zbJpMaNGaD7V3QRqhfuHpk/N30DHet3pX/Td
6OJEQI1GMN3Yi32eE3DzPk.FR1:17710:0:99999:7:::
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ben
drwxr-xr-x 2 root root 4096 Feb  8 10:09 /home/Ben
root@CS6238:/home/cs6238/Desktop/2FA# python 2FA.py
Account Mangement Tool

(1) Create User Account
(2) Login
(3) Update Password
(4) Delete User Account

Your Selection: 4
Enter Username: Ben
Enter Password for the Ben: Ben@123456
Enter Token: XRcyVF5J5jYa/YHCYoKdl.
RUNNING TOKEN GENERATOR, CHOOSE OPTION 3
Enter '1' for registering new User or '2' for generating token for existing user
or '3' for deleting user account: 3
Enter the user id:Ben
Enter PIN for the Ben :123456
hEzIy//8/B3dgclJhVzAQ.
Enter 'Y' or 'y' whether task completed successfully: y
SUCCESS: Ben deleted
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/passwd
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/shadow
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ben
ls: cannot access '~Ben': No such file or directory
root@CS6238:/home/cs6238/Desktop/2FA#
```

III. Task2: Security Analysis of 2FA

This section answers security related topics for the 2FA method implemented in Task1.

1. Discuss the advantages (any 2), disadvantages (any 2) and the possible attacks (any 2) on the 2FA method.

The 2FA method utilized for this project is has the following advantages:

- **Offers improved security against victim impersonation:** By requiring a second form of identification, 2FA decreases the probability that an attacker can impersonate a user to gain access to the victim's resources, i.e., accounts or other sensitive resources.
- **Offers improved security against fraudulent access:** By requiring a second form of identification, 2FA increases the work factor for an attacker to gain access to the victim's resources, i.e., accounts or other sensitive resources. Given this extra work, the attacker may target a victim that does not utilize 2FA.

The 2FA method utilized for this project is has the following disadvantages:

- **User acceptability:** Employing a second factor imposes more work to the user. Thus, a user may dismiss some aspects of the organization's security policy to decrease their work factor. This compromise to the security policy could even extend to non 2FA related measures.
- **Additional workload placed on the organization resources:** There will be additional work placed on an organization to implement and maintain 2FA. This includes more work to maintain the gold standard of security (authentication, authorization, audit). The 2FA method will require more resources: staffing and equipment for implementation. In addition, additional staffing (help desk, information technology and security analysts) will be necessary for maintenance of 2FA.

The 2FA method utilized for this project is susceptible to the following attacks:

- **User acceptability:** The 2FA method used for this project is vulnerable to user acceptability. For example, a user may store a PIN/TOKEN in clear text, on paper, etc., such that an attacker can easily steal the second factor.
- **PIN refresh:** The 2FA method does not require a refresh of the user's PIN. This can allow an attacker to utilize a stolen PIN to compromise the second factor for the user.
- Two vulnerabilities associated with web-based 2FA applications:
 - **A proxy-based phishing attack:** If an attacker's phishing website functions as proxy, then requests from the victim to legitimate websites can deliver responses back to the attacker's proxy. In this manner, the attacker can obtain the active session token for the victim. This session token, usually a cookie, can then be used to by the attacker to access the victim's account.
 - **Social engineering, SIM swapping:** This vulnerability relies on an attacker convincing a mobile phone Service Provider (SP) to port the victim's telephone number to a different Subscriber Identity Module (SIM), i.e., the attacker's device. To convince the SP, the attacker impersonates the victim by obtaining the victim's personal information through some nefarious method, e.g., social engineering or phishing.

2. Let us say 2FA has to be implemented in a realistic environment. Recommend one improvement for the current 2FA scheme.

The 2FA method can be improved by not requiring the user to affirm ('Y' or 'y') the token generated by the Token Generator (TG). If the user were to enter any keystroke other than the affirmation set, then the TG discards the token. This can leave the user's account vulnerable, as the token may never be updated. Thus, a token stolen by an attacker may be used for (repeated) attacks.

3. How can one implement the 2FA scheme in a server-client setting, and how will you secure the token transfer between the separate systems?
 - The 2FA scheme can be implemented by synchronizing a PIN+TOKEN between a client and an authentication server. In this application, the TOKEN is a randomly generated sequence of characters.

The combination of [something you know (PIN), something you have (TOKEN)] are combined by the user and submitted to the authentication server. The authentication server knows the user's PIN and both the user and server TOKEN generators are synchronized to generate new tokens periodically (~30 seconds). When a user is authenticated by PIN+TOKEN, the authentication server provides a secure cookie (token) back to the client.

- The token transfer for the 2FA method can be secured by employing public-key cryptography. This method would use a pair of keys: a public key that is sharable and a private key known only to the client.

I. APPENDIX

This section contains additional test cases that supplement the test cases discussed in Task 1.

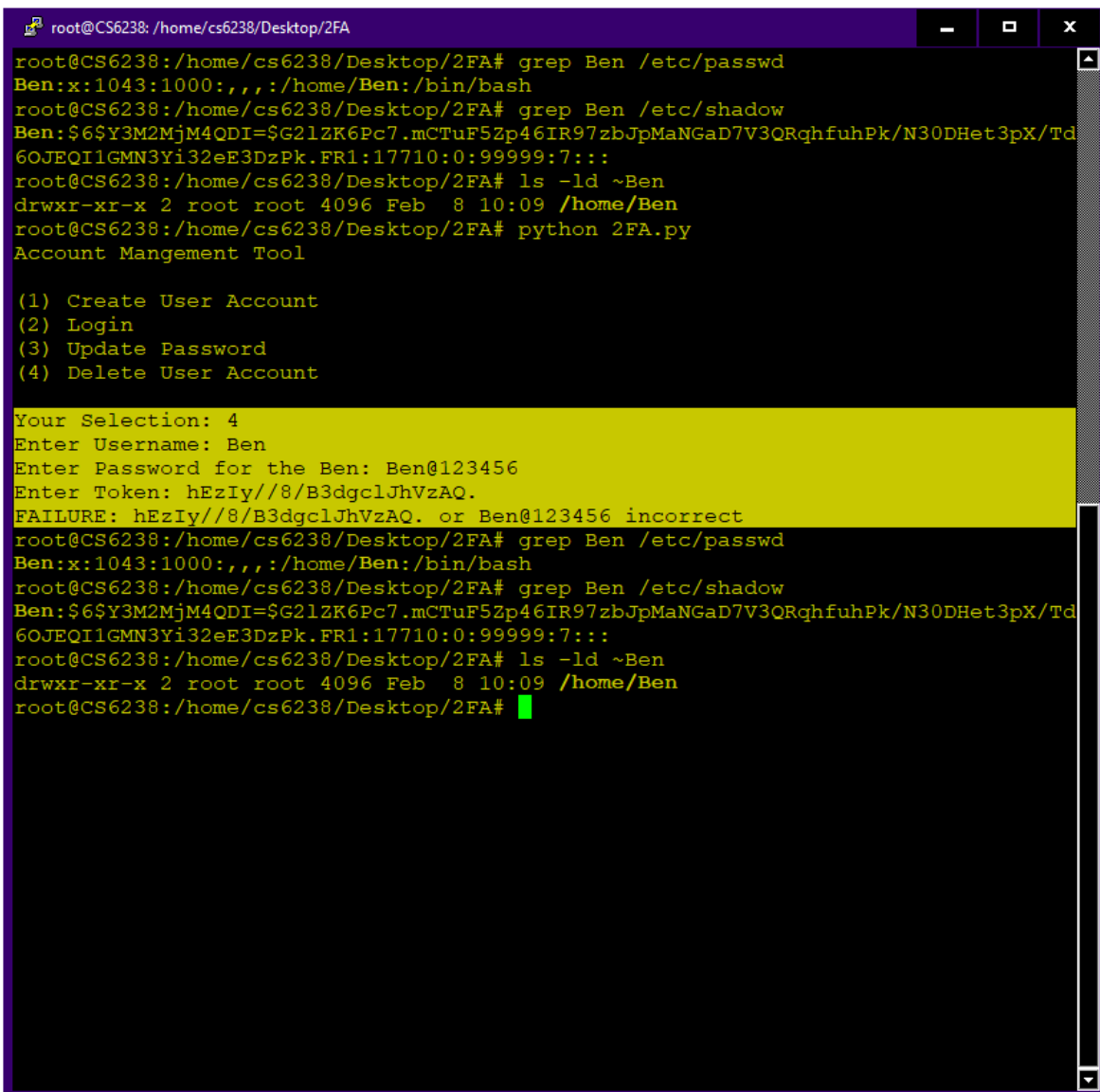
Test Case A1: Recall from Test Case 8, that the update for Ben failed due to an invalid token. Thus, running Test Case 10 with the next token will fail, i.e., Ben's login credentials are not valid because Test Case 8 failed to update the passwd with next token.

Output from Test Case 8:

```
Enter Username: Ben
Enter Password for the Ben: Ben@123456
Enter Token: hEzIy//8/B3dgclJhVzAQ.
FAILURE: hEzIy//8/B3dgclJhVzAQ. or Ben@123456 incorrect
```

Output from Test Case 10:

```
Your Selection: 4
Enter Username: Ben
Enter Password for the Ben: Ben@123456
Enter Token: hEzIy//8/B3dgclJhVzAQ.
FAILURE: hEzIy//8/B3dgclJhVzAQ. or Ben@123456 incorrect
```



```
root@CS6238: /home/cs6238/Desktop/2FA#
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/passwd
Ben:x:1043:1000:,,,:/home/Ben:/bin/bash
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/shadow
Ben:$6$Y3M2MjM4QDI=$G2lZK6Pc7.mCTuF5Zp46IR97zbJpMaNGaD7V3QRqhfuHpk/N30DHet3pX/Td6OJEQI1GMN3Yi32eE3DzPk.FR1:17710:0:99999:7:::
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ben
drwxr-xr-x 2 root root 4096 Feb  8 10:09 /home/Ben
root@CS6238:/home/cs6238/Desktop/2FA# python 2FA.py
Account Mangement Tool

(1) Create User Account
(2) Login
(3) Update Password
(4) Delete User Account

Your Selection: 4
Enter Username: Ben
Enter Password for the Ben: Ben@123456
Enter Token: hEzIy//8/B3dgclJhVzAQ.
FAILURE: hEzIy//8/B3dgclJhVzAQ. or Ben@123456 incorrect
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/passwd
Ben:x:1043:1000:,,,:/home/Ben:/bin/bash
root@CS6238:/home/cs6238/Desktop/2FA# grep Ben /etc/shadow
Ben:$6$Y3M2MjM4QDI=$G2lZK6Pc7.mCTuF5Zp46IR97zbJpMaNGaD7V3QRqhfuHpk/N30DHet3pX/Td6OJEQI1GMN3Yi32eE3DzPk.FR1:17710:0:99999:7:::
root@CS6238:/home/cs6238/Desktop/2FA# ls -ld ~Ben
drwxr-xr-x 2 root root 4096 Feb  8 10:09 /home/Ben
root@CS6238:/home/cs6238/Desktop/2FA#
```