CS2107 Assignment 2 Brief

Capture the Flag: Basic Unix Command and Practical System

Last Updated: 15 March 2018

Contents

Introduction

This assignment takes the form of an information security capture-the-flag (CTF) style competition. In a CTF, participants solve problems involving security weaknesses to bypass defences to obtain a sensitive piece of information called the 'flag'.

In this assignment, participants are exposed to some of the common skills required to play in these competitions.

Instructions

Grading Scheme and Due Date

This is an individual assignment. You are allowed to post questions on the IVLE forum but please ensure that the questions do not ask for the solution. Additionally, please do not post the answers to the challenges. When answering questions, please exercise some restraint.

This assignment is worth 26 out of 60 points allocated to all three assignments and 8.5% of the grade for the entire module. The assignment is divided into three sections:

1. Section A: Setup and Basic Tasks - 8 Points
2. Section B: Basic Challenges - 18 Points

The assignment is due 4 April 2018, 2359 HRS.

Rules and Guidelines

PLEASE READ THE FOLLOWING BEFORE BEGINNING

1. Do not attack any infrastructure not **explicitly authorised** in this document.
2. Work individually. Discussion on the forum is allowed but please refrain from posting solutions.
3. The skills taught in this assignment are not to be used on any system you do not own or have express permission to test. This is a **criminal offence** under the Singapore Computer Misuse and Cybersecurity Act.
4. All challenges have a solution. They are guaranteed to be solvable with assistance of the internet.
5. Ask the TAs for assistance only after you have exhausted every other avenue of self-help.
6. Every challenge will contain a flag and will provide the accepted flag format. Please ensure your submissions meet the flag format stated **exactly**. This means include the `flag{}` portion if the flag format asks for it.
7. Complete all the challenges before attempting the quiz to submit your flags on IVLE. Be very careful please, only one attempt will be allowed. It is recommended that you note down all your flags in a separate document.

8. The challenges are tested from the NUS WiFi within the School of Computing and outside of NUS. Connectivity cannot be guaranteed anywhere else in NUS.

One of the most important skills in the information security field is the skill of seeking an answer independently. As the assignment progresses, fewer hints will be given and it is expected that the participant be able to utilise resources discovered through Google or any other search engine to achieve the tasks.

While the challenges might not be covered in class, the topics in the assignment are very applicable to security problems in real life. In the long run, the practical skills gained would benefit participants immensely.

You may find the files you need in the package distributed with this assignment.

Contact

Please direct any inquiries about the assignment to leeyc@comp.nus.edu.sg

Acknowledgements

This assignment is a collective work of present and past teaching assistants, including Lee Yu Choy(AY 17/18), Nikolas Tay(AY 16/17), Jeremy Heng(AY 16/17) and all the online resources

Section A: Setup and Basic Tasks

The challenges in this section are meant to give the student a rough introduction to the skills required in this assignment. While the challenges in this section provide a great amount of guidance, it is expected for the student to do some measure of independent research to solve the problems.

The problems in this section are worth 0.5 points each. Except for 5.3 which is 1 point

Linux

A Linux system is crucial for solving some of the challenges, the challenges in this section will prepare you for the more advanced sections by presenting some elementary tasks to solve.

It is expected that the participant has rudimentary proficiency in using a Linux system that can be gleaned by reading the tutorial at this link: https://www.digitalocean.com/community/tutorials/an-introduction-to-the-linux-terminal. More knowledge might be needed however and it is expected that the participant do some self-exploration.

A.1.1 Creating a Linux Virtual Machine and Running an ELF Binary

`Flag Format: cs2107{<32 lowercase hexadecimal characters>}`

Files:

1. task

The objective of this task is to run the ELF binary file in a Linux terminal. There are multiple methods to achieve this but it is suggested that the following steps be followed: (Skip to Step 4 if you have installed the Virtualbox in assignment 1)

1. Install Virtualbox (https://www.virtualbox.org/wiki/Downloads)
2. Create an Ubuntu Virtual Machine (https://www.ubuntu.com/download/desktop) do get the 64bits version

3. Move the entire assignment package onto the virtual machine using shared folders
4. Navigate to the location of the `task` file in the corresponding assignment directory in the terminal
5. Adjust the permissions as necessary (chmod)
6. Run the binary and observe the output. Hint: A binary not in the standard path and in the current directory may be run by entering `./<binary_name>`.

A.1.2 Decoding a File Encoded in Base64

`Flag Format: cs2107{<32 lowercase hexadecimal characters>}`

Files:

1. decodeme.txt

The file provided is a large text file containing a base64 encoded string. Your task is to use the `base64` utility that ships with a standard installation of Linux to decode this file and look for the flag.

Hint: Try `base64 -d`.

A.1.3 Opening an Interactive Socket Connection

`Flag Format: cs2107{<32 lowercase hexadecimal characters>}`

Open a TCP socket to the following address `cs2107.codecla.ws:1337` and submit the password `epic_cat`. There are multiple ways to do this but it is recommended you use the `netcat` utility that comes with a standard installation of Ubuntu.

Hint: Try using `nc <hostname> <port>`.

A.1.4 Using One-Liner Scripts on the Command Line

`Flag Format: cs2107{<32 lowercase hexadecimal characters>}`

Often, you would find the need to do quick computation or processing on the command line that would be served by the use of scripts. A good way of integrating scripting languages into your workflow is to use one liner commands. In python, then can be activated by using the `-c` switch which allows you to pass a python script as a single line to the python interpreter.

Please run the following command to obtain the flag for this task:

```
python -c 'import md5; seed = 13371338;
print "Please see:\ncs2107{%s}" % md5.md5(str(seed)).hexdigest()' | grep cs2107
```

Forensics

This section deals with the analysis of files and their metadata.

A.2.1 Analysis of the Filetype of Multiple Files

`Flag Format: cs2107{<32 lowercase hexadecimal characters>}`

Files:

1. filetype_analysis.tar.gz

In this task, you are required to analyse the filetypes of the files and check the metadata to obtain the flag. The files are contained in a tar archive. You will need to extract the archive to get all the files using `tar`. The flag is contained in a JPEG file. You may use tools such as `file` and `exiftool` to accomplish this.

A.2.2 Analysis of the Contents of Multiple Text Files

`Flag Format: flag{<32 lowercase hexadecimal characters>}`

Files:

1. textfiles.tar.gz

In this task, you are given many text files. Your task is to find the flag contained in the text file with the value `14e9f9a13ea3426e3023`. You may wish to use a tool such as `grep` to easily search all of the files for the desired value.

A.2.3 Fixing the Binary Content of an Image File

`Flag Format: cs2107{<32 lowercase hexadecimal characters>}`

Files:

1. broken.jpg
2. notbroken.jpg

This task requires you to fix a broken JPEG file with a hex editor. You may use the unmutilated sample of a JPEG file as reference when beginning to fix the broken JPEG containing the flag.

Hint: Use the Bless hex editor. Also, hexadecimal characters do not contain the character 'O'.

A.2.4 Extracting Hidden Data with Foremost

`Flag Format: flag{<32 lowercase hexadecimal characters>}`

Files:

1. hidden.png

There is a hidden file in this PNG. Your task is to discover the hidden file and extract the flag. If you ever need a password, it is 'giveflag'. You may use the `foremost` and `unzip` tools to help you retrieve this file.

Hint: Look up file carving.

A.2.5 Wrong Command using DD

`Flag Format: cs2107{<32 lowercase hexadecimal characters>}`

Files: 1. broken.png

Bob tried to copy his file using dd command but accidentally input an extra parameters. Try to find the parameters and fix his file for him.

Hint: man dd

Web

The web is a large topic in information security and it is important to know how to interact with it in the workflow of an information security professional.

A.3.1 Using Curl to Get the Contents of a Webpage

`Flag Format: cs2107{<32 lowercase hexadecimal characters>}`

Sometimes it is useful to download the contents of a webpage or a file hosted on a web server on the command line. `curl` is one such program that can help you do that. Please grab the contents of the URL `http://cs2107.codecla.ws:8000/flag.html`.

A.3.2 Using Curl to Make POST Requests

`Flag Format: cs2107{<32 lowercase hexadecimal characters>}`

If you did not already realise, you made a GET request to the web server in the previous challenge. Those are not the only type of requests that exist. Another type of request is the POST request. Please figure out the right parameters and how to make such a request to the web application at `cs2107.codecla.ws:8001`.

A.3.3 Gaining Information Using robots.txt

`Flag Format: cs2107{<32 lowercase hexadecimal characters>}`

Web crawlers are insatiable. One way to prevent spiders such as Google or Yahoo from indexing is through the Robots Exclusion Standard. However, some web administrators include sensitive information such as paths to admin pages or secret debugging pages. It is often useful to take a look at the `robots.txt` file when performing reconnaisance on a target. Please try to find the secret file at `cs2107.codecla.ws:8002`.

Cryptography

Much of security involves the use of cryptography to keep sensitive information secret. Thus, an amount of practical cryptography knowledge is very useful.

A.4.1 Installing John the Ripper and Cracking a Hash

`Flag Format: lowercase characters word`

Files:

1. shadow

`/etc/shadow` files on Linux contain the password hashes of the users for authentication during login. We have provided a sample shadow file. Your task is to obtain the password to the admin account. To achieve this, you may use John the Ripper to crack the password in the shadow file.

A.4.2 Decrypting a One Byte XOR Scheme

`Flag Format: cs2107{<32 lowercase hexadecimal characters>}`

Files:

1. xorscheme.encrypt

The exclusive OR (XOR) operation is used extensively in cryptography operations. In this challenge, you are required to decrypt a file that has been encrypted with a one byte XOR key.

A.4.3 Decrypting a Simple Transposition Cipher [1 mark]

`Flag Format: cs2107{<32 lowercase hexadecimal characters>}`

Files:

1. transposition.encrypt

The provided file is encrypted with the Caesar's Box cipher. Please decrypt it.

Hint: Try writing the encrypted string out in a square.

Section B: Basic Challenges

The challenges in this section are a little more challenging with fewer hints given. Some of these challenges require a little scripting and quite a bit of thinking.

The problems in this section are worth 1 points each unless otherwise mentioned in the question

Linux

B.1.1 Providing Proof of Work

`Flag Format: cs2107{<32 lowercase hexadecimal characters>}`

Often, before a service allows you to use it, you have to demonstrate you are not going to abuse it by providing a 'proof of work'. In this task, you will have to calculate the hashes of 500 randomised byte strings programmatically.

Please connect to `nc cs2107.codecla.ws 9000`.

B.1.2 Exploit a Vulnerable Python Script [2 points]

`Flag Format: cs2107{<32 lowercase hexadecimal characters>}`

Files:

1. snek.py

In this task, you have to compromise a python application to retrieve the flag. This is an introduction to doing source code review for remote vulnerabilities. You are given the source so that you can examine exactly what happens during the execution of the program and to debug it locally.

You may find the script by connecting with `nc cs2107.codecla.ws 9001`.

B.1.3 Reverse Engineer a License Key from a C Binary Given Source [2 points]

`Flag Format: cs2107{[a-zA-Z0-9_]+}`

Files:

1. reverseme.c
2. reverseme

In this task, you are given a C file as well as a Linux binary. Please analyse the source code and figure out an argument satisfying the constraints found in the binary.

B.1.4 Scripting your way through Zipfile [2 points]

`Flag Format: cs2107{<32 lowercase hexadecimal characters>}`

Files:

1. zip_files.tar
2. pattern.txt("-" refer to password that you do not need to guess, e.g. -000-0- the password can be -210-7-)

In this task, you are suppose to practice some basic scripting to "brute force" a password locked zipfile.

The first zip file is "begin.zip" with the password "cs2107"

This task shows you that how ascii characters as password can be crack easily if the length is small

Hint: If you feel that you are taking too long to brute force the password(e.g. you are using all ascii characters) try to match the pattern.

Forensics

B.2.1 Analyse a JPEG Image [2 points]

`Flag Format: cs2107{[a-zA-Z0-9_]+}`

Files:

1. haiku.jpg

There is a secret message hidden in this JPEG. Reading some poetry may help you find it.

Hint: How can you hide text secrets in a JPEG file? Think a little simpler than steganography.

B.2.2 Analyse a GIF Image

`Flag Format: cs2107{[a-zA-Z0-9_]+}`

Files:

1. ghost.gif

This spooky GIF was found in the hard drive of a convicted serial killer at the scene of a brutal murder. Our investigators have revealed through tough interrogation that the serial killer likes to leave calling cards in his digital footprint.

Please find the message he left us.

Hint: Again, please be very careful when copying the flag. There are no capital 'O's in the flag.

B.2.3 Analyse a Large Data Blob [2 points]

`Flag Format: cs2107{[a-zA-Z0-9_]+}`

Files:

1. data.blob

We found this large blob of data but we can't figure out what it is. Would you kindly help us figure out what is real and what is not?

Web

B.3.1 Bypassing Checks on a Web Application

`Flag Format: cs2107{<32 lowercase hexadecimal characters>}`

Web security is also a big area to test and it would not be a CTF without web tasks. A python flask application is running on `cs2107.codecla.ws:9080`. Please try and get the flag from the server.

Cryptography

B.4.1 Decrypting Encrypted Data [2 points]

`Flag Format: cs2107{[a-zA-Z0-9_]+}`

Files:

1. secrets.enc

This file is encrypted with 256 bit AES with the key 'evilpassword'. Please find the flag.

Hint: If you're stuck, perhaps you should look up some cool people named Ron, Adi, and Leonard.

B.4.2 Decrypting an XOR Scheme

`Flag Format: cs2107{<32 lowercase hexadecimal characters>}`

Files:

1. encrypted.xor

So far, we have seen challenges involving very simple XOR operations. In this challenge, we have a message that has been encrypted with a key of unknown length. Please recover the message.

B.4.3 Mutiple Public Keys [2 points]

`Flag Format: cs2107{<32 lowercase hexadecimal characters>}`

Files:

1. enc

John heard that you know how to encrypt a file using public key now, so he encrypted his file using two public key to feel more secure. Show him that two public keys will not necessarily improve the security.

Hint: Figure out how to make use of all public key(s) before trying to find the private key.

Hint 2: After deciding the value of e, John did an incremental approach(start from a small value) to find the value of d. (Do not brute force from 1 to $\sqrt{e}$)

Conclusions

I hope you enjoyed the assignment and have learnt something new. Again, please make sure that your flags are correct and contain the flag format **EXACTLY** as stated. This includes the `cs2107{}` or `flag{}` tags.

If you have any question, feel free to post on the forum or email the assignment team directly. In addition you can just ask the oracle :)

Best regards, CS2107 assignment team

Versioning and Changelog

Current Version: v3.02

- 24/2/2018 (v3.01) - Initial Document
- 15/3/2018 (v3.02) - Added some hints