

Project 1 - Part 2: Stack Buffer Exploit

The goals of this project:

- Execute a stack buffer overflow exploit
- Understand how stack buffer overflows work
- Understand how stack registers are manipulated during program execution

The final deliverables:

The submitted paper must adhere to the following format

- Written answers to each individual question are limited to a maximum of 200 words
- Font: Times New Roman, Size 11.
- Spacing: Single Spaced with standard 1" margins
- Heading: students' gt_user_id above page number.
- Submission Format: pdf
- You are allowed to submit the paper in [JDF \(Joyner Document Format\)](#)

The following must be submitted to the Module: **Project 1 - Part 2: Stack Buffer Exploit**

- Submission name format: **gt_user_id_data.txt**
 - Example: **ctaylor308_data.txt**

The following must be submitted to the Module: **Project 1 - Part 2: Exploit Explanation**

- Submission name format: **gt_user_id_explanation.pdf**
 - Example: **ctaylor308_explanation.pdf**

Important Note: We know Canvas adds -1. That is okay.

Information:

- **Plagiarism will not be tolerated!** For information: [GaTech Academic Honor Code](#) and syllabus..
- **Papers must be cited in [IEEE format](#).** For information: Course Syllabus

Tools Needed:

You **MUST** use the latest version of VirtualBox, downloaded from: [VirtualBox](#)

- If you need the VM: [VM Download Link](#)
- GDB command cheat sheet: [Cheat Sheet](#)

Submission:

Late Work Will Be Penalized or Not Accepted

Please refer to the course syllabus for more information about submitting, deadlines, and deductions.

Project Tasks (25 points):

Download, Compile, and Run:

- Download **exploit.c** from [here](#).
- Open **exploit.c** and change the `gid` to your own within the **main()** function.
- Compile with: **gcc exploit.c -o exploit -fno-stack-protector**
- Create a .txt file with your `gid`: **gid_data.txt** and insert some values.
- Run with the following: **./exploit gid_data.txt**

You will notice that **exploit.c** sorts the information included in **gid_data.txt**

Turn off ASLR - (3 points):

For this exploit to work correctly, you will need to figure out how to turn off ASLR.

Once you figure out how to turn off ASLR, make sure to include how you figured it out in your **gid_explanation.pdf**.

Finding the Addresses - (9 points):

For this project we will need to find three addresses using GDB (Feel free to view the GDB cheat sheet). Be sure to include how you found each address in your **gid_explanation.pdf**.

Those three addresses are:

- **system()**, **"/bin/sh"**, and any **exit()** function that exits the shell cleanly.
- Include the addresses you found in this section of the write up.



Important Note:

You are **NOT** allowed to use environment variables to store these addresses.

Figuring out Padding- (3 points):

Just putting those three addresses in the **gid_data.txt** is not enough for this return-to-libc exploit to work. There is a required amount of padding that goes in this file before the exploit works. Include the explanation for the following in your **gid_explanation.pdf**.

1. How many bytes are needed for the padding.
2. How you found it.
3. Why that many were needed.

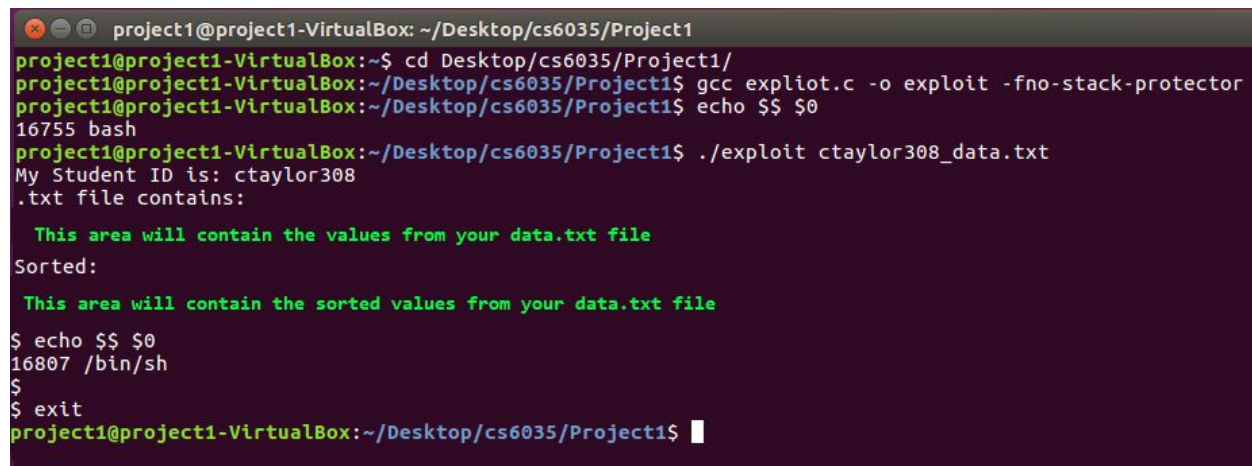
Return-to-libc using *gtid_data.txt* - (10 points):

Once all aspects of this project have been figured out, craft your *gtid_data.txt* in such a way that when you run `./exploit gtid_data.txt` a shell spawns. An example shown below. Be sure to include a screenshot of your working exploit in your **gtid_explanation.pdf**. An example:

Important: You must run the exploit from the following path: `~/Desktop/cs6035/Project1/` (You will automatically lose 100% if you do not) [You will have to create these folders]

An example execution:

1. `./Desktop/cs6035/Project1/exploit ctaylor308_data.txt`
- Or
2. `cd ~/Desktop/cs6035/Project1/`
`./exploit ctaylor308_data.txt`



```
project1@project1-VirtualBox: ~/Desktop/cs6035/Project1
project1@project1-VirtualBox:~$ cd Desktop/cs6035/Project1/
project1@project1-VirtualBox:~/Desktop/cs6035/Project1$ gcc expliot.c -o exploit -fno-stack-protector
project1@project1-VirtualBox:~/Desktop/cs6035/Project1$ echo $$ $0
16755 bash
project1@project1-VirtualBox:~/Desktop/cs6035/Project1$ ./exploit ctaylor308_data.txt
My Student ID is: ctaylor308
.txt file contains:

  This area will contain the values from your data.txt file
Sorted:
  This area will contain the sorted values from your data.txt file
$ echo $$ $0
16807 /bin/sh
$
$ exit
project1@project1-VirtualBox:~/Desktop/cs6035/Project1$
```

Important Notes:

- Your terminal screenshot should contain the same commands as the one above
- Your terminal screenshot should be executed from the **correct** folder path
- Your terminal should show the values both before and after sorting
- No segfaults are allowed. This must execute a clean shell, then exit cleanly.
- In the screenshot, it shows the exploit working for **ctaylor308_data.txt** but in your screenshot it should be your **gtid_data.txt**

Deductions:

- - 5 Points: No Screenshot shown (Show a screenshot, even if it is a failed attempt)
- - 2 Points: Correct echo commands not shown
- - 2 Points: Incorrectly named data.txt file
- - 2 Points: Did not put your gtid.
- - 2 Points: Incorrectly named exploit.c file
- - 10 Points: Folder path is incorrect
- - 5 points: Clean exit is not achieved