Online 1 - A2

You are given a vulnerable C program named **`stack-A2.c`**. Replace `<PARAM_1>`, `<PARAM_2>`, `<PARAM_3>` in the source code with the corresponding values of the table.

Tasks

- First, compile the program ***for 64 bit*** from the root's privilege and set its UID as shown in the lab. Do not forget to turn off address space randomization and stack protection. Also, make sure that the stack is executable while compiling the program.
- Prepare a payload (e.g. badfile) which will cause the program to open a shell with root's privilege when executed by other users (i.e., seed).
- Assume that you only have a **dash** shell. So you cannot link the default shell to **zsh**. To accomplish this, run

  ```
  sudo ln -sf /bin/dash /bin/sh
  ```

  instead of

  ```
  sudo ln -sf /bin/zsh /bin/sh
  ```

  before you start.
- Expected Output:

  ```
  In main function
  What do you think?
  sorry, hack failed!!
  # whoami
  root
  #
  ```

- Make sure that you don't change the C program other than the macro parameters values as instructed.
- If you have used cloud VM, make sure to write the public ip of the vm as a comment in the `exploit.py` file.
- Rename your exploit.py file with 18050xx.py and submit in moodle.

	PARAM_1	PARAM_2	PARAM_3
1805031	207	511	817
1805032	220	528	836
1805033	233	545	855
1805034	246	562	874
1805035	259	579	893
1805036	272	596	912
1805037	285	613	931
1805038	298	630	950

1805039	311	647	969
1805040	324	664	988
1805041	337	681	1007
1805042	350	698	1026
1805043	363	715	1045
1805044	376	732	1064
1805045	389	749	1083
1805046	402	766	1102
1805047	415	783	1121
1805048	428	800	1140
1805049	441	817	1159
1805050	454	834	1178
1805051	467	851	1197
1805052	480	868	1216
1805053	493	885	1235
1805054	506	902	1254
1805055	519	919	1273
1805056	532	936	1292
1805057	545	953	1311
1805058	558	970	1330
1805059	571	987	1349
1805060	584	1004	1368
	597	1021	1387
	610	1038	1406