

UP23 Lab02

Date: 2023-03-13

- UP23 Lab02
- Find a Needle in a Haystack
 - The Challenge Server
 - Uploading Your Solver
 - Grading

Find a Needle in a Haystack

This lab aims to practice implementing a directory traversal program that finds a file containing a specific magic number from a target directory and its sub-directories.

The Challenge Server

The challenge server can be accessed using the `nc` command:

```
nc up23.zoolab.org 10081
```

Upon connecting to the challenge server, you must first solve the Proof-of-Work challenge (ref: pow-solver (<https://md.zoolab.org/s/EHSmQ0szV>)). Then you can follow the instructions to upload your solver implementation, which must be compiled as a static binary to solve the challenge.

Once your solver has been uploaded to our server, we will run your solver in a clean Linux runtime environment. Two additional parameters are passed to your solver. Suppose your solver is placed at `/solver`. It is invoked with the parameters:

```
/solver /path/to/the/files/directory magic-number
```

There will be at least one file containing the magic number (in ASCII). Your solver has to find one file and print out the file's full pathname to the standard output (`stdout`).

Our challenge server checks the filename output from your solver's `stdout` and verifies if the solver has found the correct file.

Note that the challenge server only checks the `stdout` from your solver. If you want to debug your program when running on the challenge server, you can output your debug message to `stderr`.

Uploading Your Solver

To simplify the uploading process, you can use our provided `pwntools` python script to solve the pow and upload your solver binary executable. The upload script is available here (view (<https://up23.zoolab.org/code.html?file=up23/lab02/submit.py>) | download (<https://up23.zoolab.org/up23/lab02/submit.py>)). You have to place the `pow.py` file in the same directory and invoke the script by passing the path of your solver as the first parameter to the submission script.

Your solver must be compiled with the `-static` option.

Grading

- [10 pts] You can successfully upload your solver to the challenge server and pass an arbitrary pathname to the challenge server via `stdout`
- [85 pts] Your solver can find the correct file that contains the magic numbers requested by the challenge server.

We have an execution time limit for your challenge. You have to solve the challenge within 60s.