Stack Overflow Problem Set

Description

The goal of this problem set is to perform a basic stack overflow exploit against a vulnerable program, and develop a patch to remove the vulnerability.

To complete the problem set, you will need to ssh to your container at `$user@amplifier.ccs.neu.edu:$port`, where `$user` is your gitlab username and `$port` is your assigned ssh port (https://seclab-devel.ccs.neu.edu/snippets/6). Authentication is performed using any of your uploaded ssh public keys in gitlab.

You will also need to clone the problem set repository located at `git@seclab-devel.ccs.neu.edu:softvulnsec/prset01.git`.

Important Information	
Available	Tue 10 Feb 10:00 EST
Submission Deadline	Tue 17 Feb 18:00 EST
Gitlab URL	https://seclab-devel.ccs.neu.edu/softvulnsec/prset01 (https://seclab-devel.ccs.neu.edu/softvulnsec/prset01)

Vulnerability Identification

A compiled version of the vulnerable program from `prset01.git` is located at `/usr/local/bin/prset01` on your container. Identify the address on the stack of a saved return address that can be overwritten by examining the source code and running the program in gdb. Additionally, identify the address of a buffer you can overflow on the stack.

Overflow Length

Compute the length between the start of the buffer and the location of the saved return address on the stack.

Controlling the IP

Inject a payload into the program that redirects control flow to your buffer. Convince yourself that this occurs by either tracing execution using gdb, filling the buffer with instructions that trap (e.g., `int3` (`0xcc` bytes)), or some other method.

Obtain a Secret

Notice that the vulnerable program is setuid 1001. Write an exploit payload that leaks the contents of `/usr/local/share/prset01.secret`, which is only readable by UID 1001.

Feel free to adapt the example payload from gitlab or develop your own for extra points. Using a shellcode generator like metasploit is not permitted, however.

Patch the Vulnerability

Fork the repository for this problem set in gitlab – your copy should have the URL `git@seclab-devel.ccs.neu.edu:$user/prset01.git`. Develop a patch to the original source file that removes the vulnerability while preserving the intended functionality of the program.

Commit the patch to your repository (i.e., edit the original source code and commit the changed version).

Answer Submission

In your forked repository, commit a JSON object in the file `solution.json` with the following format:

```
{
    "buf_addr": "<buffer address>",
    "ret_addr": "<address of return address>",
    "payload_len": <length of payload>,
    "secret": "<secret value>"
}
```

An example solution might look like:

```
{
    "buf_addr": "0x1234",
    "ret_addr": "0xabcd",
    "payload_len": 100,
    "secret": "00000000"
}
```

You are responsible for submitting valid JSON at the correct path. Use a validator if you're unsure about this, and double-check that your JSON follows the format above exactly.

Commit the source code for your exploit script(s) and payload.

Commit a `README.md` that describes the vulnerability, your exploit code, and how your patch fixes the vulnerability.

Push all commits to gitlab.

Extra Credit

The vulnerable program permits a fairly trivial code reuse attack. Implement this attack, commit your exploit, and describe your solution in a separate section of `README.md` for extra credit.

Updated Fri 07 Aug 2015 10:38 EDT
Revision master/52d826a