CA647 Secure Programming Assignment

¶ Darragh O'Brien B.Sc. Ph.D. FHEA
School of Computing
Dublin City University

2023-24

Overview

In this assignment you will analyse some server code for vulnerabilities and remotely exploit those vulnerabilities to take over the server's host machine. You will carry out all your work on the CA647 virtual machine downloadable from this Google Drive directory. (VM usernames are *student* and *root* and corresponding passwords are *student* and *root*.)

Exploiting the server

A remote multithreaded server is listening on port 8001. You know the following:

- The server was built on and is executing on a 32-bit x86 machine running the same Linux distribution as the CA647 virtual machine,
- address space layout randomisation (ASLR) is enabled,
- the No eXecute (NX) feature is disabled,
- the server was built with `gcc` version 3 (the same version as is installed in the CA647 virtual machine) as follows:

```
gcc -mpreferred-stack-boundary=2 -march=i386 -o ca647_server ca647_server.c -lpthread
```

A client connects to the server and sends its name whereupon the server displays the date or time depending on the client's selection. Here is an example client.

Task

Exploit the vulnerable server in order to gain **remote** control of the host on which the server executes.

Suggested approach

1. Study the server to understand what it does and identify any flaws it contains.
2. Determine how those flaws can be exploited to cause the server to execute your payload.
3. Work out what the payload needs to do in order to work over the network.

4. Develop the payload.
5. Write and test the exploit.

Deliverables

On behalf of the team, **one team member** must submit a single zip file containing precisely the following (any additional submitted files will be ignored):

- Your commented payload (in ATT assembly format) in **payload.s**.
- Your commented exploit program i.e. the program which injects the payload in **exploit.c**.
- A description in **description.txt** of:
  - how your attack works,
  - problems with your approach (e.g. Are there any outstanding technical issues? Is the attack easily detectable?),
  - alternative approaches to implementing the attack that you considered during the assignment and their strengths and weaknesses,
  - a list of all the issues you identified in the server's code (including ones you did not exploit).
- A **video** demonstrating your successful exploit in action (here is a sample).
- A completed Declaration of Academic Integrity for each of the assignment team members.

Learning objectives

You gain the following:

1. An appreciation of the consequences of introducing remotely exploitable vulnerabilities into your code.
2. An understanding of how remote exploits work and thus possible defences against them.
3. Reverse engineering skills useful for analysing attacks.
4. Linux system interface programming skills.
5. gdb skills.

What's it worth?

15% of your overall mark.

Do I have to work in a team?

Yes. Only submissions by teams of 3 (occasionally 4) people are acceptable.

Academic integrity

- All submissions must adhere to DCU's Academic Integrity Policy.
- Sharing your work with anyone is a breach of the above policy.
- Copying work from anyone is a breach of the above policy.
- All assignment submissions will be analysed for signs of collusion and/or copying.
- Any breach of the above policy is a serious offence that will result in penalties and/or the application of disciplinary procedures.
- All submissions must include a completed Declaration of Academic Integrity for each assignment team member.

How to submit

- Upload a single zip file containing all assignment deliverables here before the deadline.
- Note there is a 50MB upload limit so ensure your video does not cause you to exceed it.

Deadline

23:59 Sunday 26 November 2023.

Late submissions

Late submissions will not be accepted.

FAQ

Q. Can you supply a rough marking guide?
A. Yes. Here is one.

Q. Can I modify the server code?
A. Feel free to modify it during experimentation but you should aim for a finished exploit that runs against the unmodified version.

Q. Can I rely on `netcat` (`nc`)?
A. You may only use `netcat` on the client-side. Relying on `netcat` on the server-side is an obvious limitation to your approach since there is no guarantee `netcat` will always be available. A better (where better means receiving more marks) solution would be one which did not rely on any server-side software.

Q. What does it mean to gain remote control of the host?
A. It means it should execute your shell commands across the network.

Q. Can I develop my solution on my laptop/on a lab machine?
A. Where you develop your solution is up to you. However, your submitted solution to the assignment must run on the provided CA647 VM. Differences in operating system or compiler versions may influence solutions. Ensuring that your submission runs on the provided CA647 VM is your responsibility.

Q. I cannot run the VM on my laptop. How can I develop a solution?
A. Work on a lab machine.

Q. I cannot get my exploit working with the server. What can I do?
A. If you are getting nowhere, you may modify the server to send you information and/or disable address space layout randomisation to make your exploit simpler to implement. You will be rewarded for demonstrating a working exploit but will obviously lose marks for having had to simplify the task.

Q. I still cannot successfully exploit the server. What can I do?
A. Start with something simple. For example, apply the same approach as used in labs by adding code to the server to launch a shell (where the shellcode is stored in some buffer). Move on from there to execute more advanced shellcode in the same way. Finally move on to exploiting the server remotely. Remember you need *something* to demonstrate.

Q. Generating suitable payload is proving a big problem. What can I do?
A. You can use third party payload. Ensure it is fully commented and referenced.

Q. How many vulnerabilities does the server contain?
A. At least one. Possibly more.

Q. A video is required. How long should it last?
A. No longer than 2 minutes.

Q. What should appear in the video?
A. The video must show a successful exploit in action against the server (with address space layout randomisation enabled). A sample is provided above.

Q. How can I generate the video?
A. In any way you wish e.g. using `recordmydesktop`, using VirtualBox's video capture facility, etc.

Q. What video encoding should I use?
A. I do not mind. The video must however be playable using VLC on a Linux lab machine in the School of Computing. It is your responsibility to test it.

Q. How do I enable/disable address space layout randomisation?
A. See the instructions in lab 1.