Secure Software

Project 2 : Application flaws

Pluquet Frédéric (FPL)

We ask you to find and exploit the flaws of the reference application specially developed for this project.

Step 0

You need a linux environment (as Debian) and tools as gcc, python, gdb, hexedit, hexdump, objdump, … You can use VirtualBox with an existing image (use Debian 8.9.0 Jessie for good VirtualBox tools integration).

Download application binary here : https://www.dropbox.com/s/ejkoa244v37muu5/project?dl=0 This is a program written in C, complied with gcc 4.9.2 with options `-m32 -fno-stack-protector -z execstack -g0`.

Step 1 (2 pts)

A secret string must be found in this binary (when you see it, you know you found it :D). Explain me (with screenshots) how you find it.

Step 2 (5 pts)

Study the assembler. Find the buffer overflow and how to exploit it to pass the password protection[1]. You will get an url.

Optional : Find the correct password (1 pt bonus).

Step 3 (10 pts)

Download the second program, compiled in same conditions. Set the setuid bit to be executed as root. Use this program (and its vulnerability) to introduce a shell code allowing you to have a shell as root on the machine.

Step 4 (3 pts)

Give me a personal and interesting example of usage of canaries to protect data against buffer overflow (and explain why).

Step 5 (2 pts bonus)

Give me a personal and interesting example of an exploit of a string format vulnerability.

[1] You must use the buffer overflow vulnerability to success this project. There are other ways (retrieve password or url with gdb, …) to get the url but it is not the purpose of this project.

Practical details

This project can be done by one student or by a well identified group of two students.

A report (in PDF format) of a minimum of pages will explain the exploit we did, with a lot of screenshots. We have to see that you understand the different vulnerabilities exploited.

This report and the code of step 4 (and step 5 if present) must be zip in one file named « SS-ID-FIRSTNAME-LASTNAME-Project-Application.zip » where you replace the "ID-FIRSTNAME-LASTNAME" by your id, firstname and lastname (« SS-ID-FIRSTNAME-LASTNAME-ID-FIRSTNAME-LASTNAME-Project-Application.zip » if you are a group of two students).

You will send this file to my Dropbox (https://www.dropbox.com/request/Hkb6gIAO0EyYEvgJBr0u) for January, 6th 2018.

References

- Global approach
  - https://www.cs.ru.nl/E.Poll/ss/slides/2_BufferOverflows.pdf
  - http://www.cs.colostate.edu/~massey/Teaching/cs356/RestrictedAccess/Slides/356lecture22.pdf
- GDB
  - https://www.cs.rochester.edu/~nelson/courses/csc_173/review/gdb.html
- Reverse Engineering
  - https://beginners.re/RE4B-EN.pdf
- Shellcode injection
  - https://dhavalkapil.com/blogs/Shellcode-Injection/
  - https://www.exploit-db.com/docs/28475.pdf
  - https://fr.slideshare.net/DhavalKapil/shellcode-injection
  - https://www.root-me.org/fr/Documentation/Applicatif/ (french)
  - https://samsclass.info/127/proj/lbuf1.htm
  - https://samsclass.info/127/proj/lbuf2.htm
  - https://samsclass.info/127/proj/p2-lbuf2.htm
- Canaries
  - http://duartes.org/gustavo/blog/post/epilogues-canaries-buffer-overflows/