# ACE Group Practical Exercises

Hello and welcome to the Aerospace Cyber Effects (ACE) Group Practical Exercise Series! This is a set of nine exercises designed to challenge your skills in programming, reverse engineering, and vulnerability analysis. Try to solve as many as you can in the time limit specified, but don’t worry if you don’t solve everything.

Thank you for trying these exercises and good luck!

## Exercises Formats

Exercises come in two formats:

* Answer a question - You are given a prompt to answer, such as “What language is this program written in?” Successfully solving the exercise means knowing the answer to the question and being able to explain how you reached that conclusion.
* Find the password (“flag”) – You are given an executable or source code file with the objective of identifying a hidden password string contained within. Successfully solving the exercise means finding the password and being able to explain how you found it.

Even if you don’t solve a problem, keep notes about what you tried and walk us through your process for trying to solve the exercise.

## Running the Exercises

We strongly recommend that you perform all analysis and execution of exercises in a Virtual Machine (VM). We recommend the 64-bit Kali Linux VM available at <https://www.kali.org/downloads/>.

Not all exercises require execution - some can be analyzed as-is / without running any code. ACE Group do not provide any hints as to which exercises require execution; identifying this is part of each exercise.

## Technical Support

If you have general questions about the exercises, run into any technical issues, or find that there’s a bug in one of the exercises preventing it from being solved, please contact Matteo Tarbet of ACE Group at <[matteo.tarbet@ngc.com](mailto:matteo.tarbet@ngc.com)>.

## Exercise Details

**Exercise 1**  
**Filename**: exercise\_01.c

There's a security problem with this code. Find it and explain how it works and how it can be exploited.

**Exercise 2**  
**Filename**: exercise\_02

Reverse engineer this program and figure out what the correct password is.

**Exercise 3**  
**Filename**: exercise\_03.py

For some reason this access control program isn't working properly. Figure out why - tell us what is wrong with it and how it could be fixed. (This is written for Python 3).

**Exercise 4**  
**Filename**: exercise\_04

We need to get started on reverse engineering this binary. What is the entry point for this executable? What is the address of main? What are the addresses of the user-defined functions called in main?

**Exercise 5**  
**Filename**: exercise\_05

Find a valid serial number and feed it to this executable! Tell us how you found a valid key - what makes it valid? Note: there are many possible solutions, but we'd recommend not resorting to brute force!

**Exercise 6**  
**Filename**: exercise\_06.txt

We found the following file on a compromised machine. We think it's related to an exploit used on the system. Can you identify what it is, what it does, and how it does it?

**Exercise 7**  
**Filename**: exercise\_07.c

The bad guys are crashing our server via some vulnerability in the attached code. Can you help figure out what the bug is?

**Exercise 8**  
**Filename**: exercise\_08

We know this program is somehow encrypting and decrypting data. Can you find the encryption keys and decrypt all of the data in this executable?

**Exercise 9**  
**Filename**: exercise\_09

We found this binary lying around but can't figure out what it does. Can you find the hidden message within?