# picoCTF2019 | Reverse Engineering Writeup

### Reverse Engineering Writeup

**Table of Contents**



---

vault-door-training - Points: 50

Your mission is to enter Dr. Evil's laboratory and retrieve the blueprints for his Doomsday Project. The laboratory is protected by a series of locked vault doors. Each door is controlled by a computer and requires a password to open. Unfortunately, our undercover agents have not been able to obtain the secret passwords for the vault doors, but one of our junior agents obtained the source code for each vault's computer! You will need to read the source code for each level to figure out what the password is for that vault door. As a warmup, we have created a replica vault in our training facility. The source code for the training vault is here: VaultDoorTraining.java

**Hint:** The password is revealed in the program's source code.

---

vault-door-1 - Points: 100

This vault uses some complicated arrays! I hope you can make sense of it, special agent. The source code for this vault is here: VaultDoor1.java

**Hint:** Look up the charAt() method online.

Below a python script that solve the challenge

---

vault-door-3 - Points: 200

This vault uses for-loops and byte arrays. The source code for this vault is here: VaultDoor3.java

**Hint:** Make a table that contains each value of the loop variables and the corresponding buffer index that it writes to.

Below a python script that solve the challenge

---

vault-door-4 - Points: 250

This vault uses ASCII encoding for the password. The source code for this vault is here: VaultDoor4.java

**Hint:** Use a search engine to find an "ASCII table"

**Hint:** You will also need to know the difference between octal, decimal, and hexademical numbers

Below a python script that solve the challenge

---

vault-door-5 - Points: 300

In the last challenge, you mastered octal (base 8), decimal (base 10), and hexadecimal (base 16) numbers, but this vault door uses a different change of base as well as URL encoding! The source code for this vault is here: VaultDoor5.java

**Hint:** You may find an encoder/decoder tool helpful, such as https://encoding.tools/

**Hint:** Read the wikipedia articles on URL encoding and base 64 encoding to understand how they work and what the results look like.

Below a python script that solve the challenge

---

vault-door-6 - Points: 350

This vault uses an XOR encryption scheme. The source code for this vault is here: VaultDoor6.java

**Hint:** If X ^ Y = Z, then Z ^ Y = X. Write a program that decrypts the flag based on this fact.

Below a python script that solve the challenge

---

vault-door-7 - Points: 400

This vault uses bit shifts to convert a password string into an array of integers. Hurry, agent, we are running out of time to stop Dr. Evil's nefarious plans! The source code for this vault is here: VaultDoor7.java

**Hint:** Use a decimal/hexademical converter such as this one: https://www.mathsisfun.com/binary-decimal-hexadecimal-converter.html

**Hint:** You will also need to consult an ASCII table such as this one: https://www.asciitable.com/

Below a python script that solve the challenge

---

asm1 - Points: 200

What does asm1(0x610) return? Submit the flag as a hexadecimal value (starting with '0x'). NOTE: Your submission for this question will NOT be in the normal flag format. Source located in the directory at /problems/asm1_1_95494d904d73b330976420bc1cd763ec.

**Hint:** assembly conditions (https://www.tutorialspoint.com/assembly_programming/assembly_conditions.htm)

---

asm2 - Points: 250

What does asm2(0x10,0x18) return? Submit the flag as a hexadecimal value (starting with '0x'). NOTE: Your submission for this question will NOT be in the normal flag format. Source located in the directory at /problems/asm2_0_a50f0b17a6f50b50a53305ebd71af535.

**Hint:** assembly conditions (https://www.tutorialspoint.com/assembly_programming/assembly_conditions.htm)

Otherwise you can use the assembly function as function of a C program in the following way

---

asm3 - Points: 300

What does asm3(0xc264bd5c,0xb5a06caa,0xad761175) return? Submit the flag as a hexadecimal value (starting with '0x'). NOTE: Your submission for this question will NOT be in the normal flag format. Source located in the directory at /problems/asm3_1_b71abaa5cc92e3f7061f23957206b434.

**Hint:** more(?) registers (https://wiki.skullsecurity.org/index.php?title=Registers)

As before it's possible to use the assembly function as function of a C program in the following way

---

asm4 - Points: 400

What will asm4("picoCTF_c1373") return? Submit the flag as a hexadecimal value (starting with '0x'). NOTE: Your submission for this question will NOT be in the normal flag format. Source located in the directory at /problems/asm4_5_ca12dca0134f6b54a52c905ffc1e5b35.

**Hint:** Treat the Array argument as a pointer

As before it's possible to use the assembly function as function of a C program in the following way

---