Basic Malware RE

Basic Malware RE

This room aims towards helping everyone learn about the basics of "Malware Reverse Engineering".

[Task 1] Introduction

These challenges are aimed towards learning about the "Static Analysis" technique used to analyze the malware. The main aim for this room is not to used any types of debuggers neither the executable's/programs should be run on any platform. You are required to answer all the questions without even using the debugger and even not executing the executable's/programs.
Meanwhile all the credits goes to @MalwareTechBlog for creating these awesome challenges.

Note:
If you have already solved these challenges - give it a try again while giving enough time to the newbies who want to learn about "Malware Analysis". Also don't try to copy paste stuff from other blogs/walkthroughs as it won't lead you to learn this amazing field. If you are having hard time solving these challenges. Study more about it and the techniques which are involved. Meanwhile you can also join TryHackMe discord and fire up you problems in there.
Password for the ZIP is MalwareTech.

#1
Read the above.

No answer needed

[Task 2] Strings :: Challenge 1

This executable prints an MD5 Hash on the screen when executed. Can you grab the exact flag?
Note: You don't need to run the executable!

#1
What is the flag of which that MD5 gets generated?

--created Ghidra project and analyzed executable
--went to md5_hash function and analyzed code to find flag

FLAG{CAN-I-MAKE-IT-ANYMORE-OBVIOUS}

[Task 3] Strings :: Challenge 2

This executable prints an MD5 Hash on the screen when executed. Can you grab the exact flag?
Note: You don't need to run the executable!

#1
What is the flag of which that MD5 gets generated?

FLAG{STACK-STRINGS-ARE-BEST-STRINGS}

--opened executable in Ghidra, went to md5_hash function, and decompiled
--decompilation shows lots of variables assigned hex values as shown below
--decoded hex values into ascii characters to get flag

```
  local_2c = 'F';
 local_2b = 0x4c;
 local_2a = 0x41;
 local_29 = 0x47;
 local_28 = 0x7b;
 local_27 = 0x53;
 local_26 = 0x54;
 local_25 = 0x41;
 local_24 = 0x43;
 local_23 = 0x4b;
 local_22 = 0x2d;
 local_21 = 0x53;
 local_20 = 0x54;
 local_1f = 0x52;
 local_1e = 0x49;
 local_1d = 0x4e;
 local_1c = 0x47;
 local_1b = 0x53;
 local_1a = 0x2d;
 local_19 = 0x41;
```

```
local_18 = 0x52;
local_17 = 0x45;
local_16 = 0x2d;
local_15 = 0x42;
local_14 = 0x45;
local_13 = 0x53;
local_12 = 0x54;
local_11 = 0x2d;
local_10 = 0x53;
local_f = 0x54;
local_e = 0x52;
local_d = 0x49;
local_c = 0x4e;
local_b = 0x47;
local_a = 0x53;
local_9 = 0x7d;
local_8 = md5_hash(&local_2c);
```

[Task 4] Strings 3 :: Challenge 3

This executable prints an MD5 Hash on the screen when executed. Can you grab the exact flag?
Note: You don't need to run the executable!

#1
What is the flag of which that MD5 gets generated?

FLAG{RESOURCES-ARE-POPULAR-FOR-MALWARE}

--opened executable in Ghidra, ran analyzer, and decompiled
--flag is shown in entry function