Name: ID:

Malware – Master on Cybersecurity

Final Exam, January 12th 2021

This exam is individual, you cannot receive any external help to perform it. Use of any external source will be punished accordingly

Don't forget to specify your full name and Identity Card number on top of this page. You **don't** need to do that for all the pages on the exam

The exam puntuation goes from 0 to 10, where 0 is no correct answer and 10 is the perfect exam

Each question has its value indicated with all the subsections' values as well

The exam may be resolved using one of the following languages:

- Catalan
- Spanish
- English

It is mandatory to explain and develop all your answers to get the full punctuation

Duration: 1 hour and 55 minutes (No extension will be granted)

Question 1 – Buffer Overflows and Injection (4 points)

Answer the following question regarding infection propagation lesson.

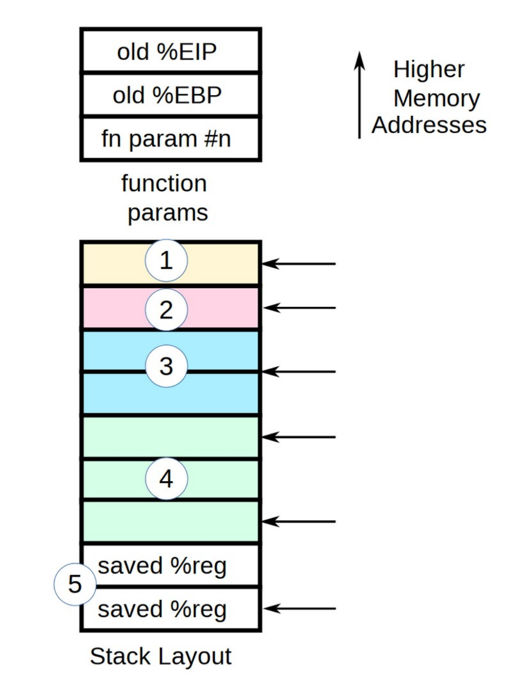
1. Given the following C code, indicate the **contents and sizes** of each field in the stack when calling the func function:

```
#include <stdio.h>
#include <string.h>

void func(char *name, int file) {
    char buffer[50];
    strcpy(buffer, name);
    printf("Welcome to our temple %s, %d\n", buffer, file);
    …
}

int main(int argc, char *argv[]) {
   func(argv[1], 7);
   return 0;
}.
```

(0.75 Points)

In next page you have the layout of the stack, just specify the number, the contents and its size. Be careful, the fields shown below not necessarily indicate their size, look at them as placeholders which can be of different size. In the solution of the problem you can indicate for example:

7. IP 64 bits.



2. Describe what are buffer overflows indicating the logic and how can be mitigated by compilers. **(0.75 Points)**

3. Describe the logic of DLL injection, how it works and its challenges. **(0.75 Points)**

4. Describe what is ROP and how can be used. **(0.75 Points)**

5. Describe the process by which a virus infects an executable using the Into Code Tail infection technique. Indicate also the advantages and limitations of this technique. **(1 Points)**

Question 2 – Obfuscation (2 points)

Answer the following question regarding obfuscation techniques lesson.

1. Given the following code:

```
0000000000401000 <_start>:
  401000:       eb 25                   jmp    401027 <ender>

0000000000401002 <starter>:
  401002:       48 31 c0                xor    %rax,%rax
  401005:       48 31 db                xor    %rbx,%rbx
  401008:       48 31 d2                xor    %rdx,%rdx
  40100b:       48 31 c9                xor    %rcx,%rcx
  40100e:       b0 01                   mov    $0x1,%al
  401010:       bf 01 00 00 00          mov    $0x1,%edi
  401015:       5e                      pop    %rsi
  401016:       ba 06 00 00 00          mov    $0x6,%edx
  40101b:       0f 05                   syscall
  40101d:       48 31 c0                xor    %rax,%rax
  401020:       b0 3c                   mov    $0x3c,%al
  401022:       48 31 ff                xor    %rdi,%rdi
  401025:       0f 05                   syscall

0000000000401027 <ender>:
  401027:       e8 d6 ff ff ff          call   401002 <starter>
  40102c:       68 65 6c 6c 6f          db     hello
  401031:       0a                      db     0xa
```

Obfuscate it using the placeholder technique studied in class. Remember the following opcodes:

`jmp → eb`

`mov rax, [LITERAL VALUE] → 48 b8` **(1 Point)**

2. Why metamorphic viruses are more powerful than polymorphic? **(0.5 Points)**

3. How can a virus know it is being run on a Virtual Machine? **(0.5 Points)**

Question 3 – Malware Categorization and Ethical Hacking (2 points)

Answer the following questions related with the malware categorization lesson.

1. Describe what is a portless backdoor and how it works. **(0.5 Points)**

2. Explain how a user-mode rootkit works and how it differentiates from a Trojan horse. **(0.5 Points)**

3. Discuss about worms, their life-cycle and how they propagate. **(0.5 Points)**

Question 4 – General Theory (2 points)

Answer the following question marking the appropriate cell. Each question has only one valid response.

Each correct answer gives 0.5 points. WRONG ANSWERS SUBTRACT 0.25 points, you can decide to leave blank answers. The minimum punctuation for the test is 0 (it doesn't affect the punctuation of other questions).

1. Regarding heap overflows:
   - □ a) Through specific input to the application they abuse memory allocation bugs and optimizations to own buffers, leading to potential arbitrary code execution.
   - □ b) By particular input to the application they affect both the heap and the stack, disrupting them and potentially leading to arbitrary code execution.
   - □ c) By overflowing the heap, using specific input to the application, they lead to application crashes and potential arbitrary code execution.

2. Disassemblers based on linear sweep
   - □ a) Are more powerful than recursive traversal
   - □ b) Are quite easy to fool because they interpret the code but not perfectly
   - □ c) Blindly decode found instructions without semantics

3. Nebbett's Shuttle:
   - □ a) It's a platform agnostic mechanism that allows to overwrite any application memory space.
   - □ b) A Win32 based mechanism which launches a process and then overwrites its memory space. There are similar approaches in Linux and other systems
   - □ c) A Win32 based mechanism for code injection on disk.

4. Which is the most critical aspect of an antivirus regarding stability?:
   - □ a) The antivirus database, as it is the block in charge of threat detection.
   - □ b) The engine, since it is the brains of the operation.
   - □ c) The engine, which is the one in charge of parsing the files on disk.