Exercise – 12 Task 2

**Task**

Hacker101 complete the “Hello World” native binary challenge

**Tools**

Ghidra, Burp Suit and Firefox´s inspect element functions

**Answer**

**Step 0:** I ran the command ‘file vulnerable’ in the terminal and found out that the binary file is ELF 64bit LSB executable.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

**Step 1**: I tried to insert 50 chars as input. This led to segmentation fault. Segmentation fault means that I try to access memory which do not belong to me.

Et bilde som inneholder tekst

Automatisk generert beskrivelseInput: “01234567890123456789012345678901234567890123456789”

Et bilde som inneholder tekst

Automatisk generert beskrivelseOutput: “Segmentation fault (core dump)”

**Step 2:** I found the amount of chars possible to insert without receiving segmentation fault. 40 chars were the limit. In the end of the output we also received some weird characters “á¢¥÷ÿ!”. I suspect this is a pointer, pointing to an address.

Et bilde som inneholder tekst

Automatisk generert beskrivelseInput: “0123456789012345678901234567890123456789”

Et bilde som inneholder tekst

Automatisk generert beskrivelseOutput: “Hello 0123456789012345678901234567890123456789á¢¥÷ÿ!”

**Step 3:** Used Ghidra to find the memory address to the flag. In Ghidra the memory addresses are stored backwards, after the Little Indian standard.

Et bilde som inneholder bord

Automatisk generert beskrivelse

Address:

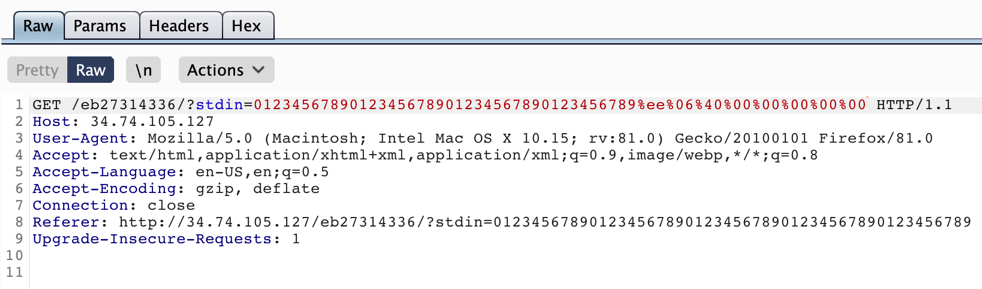
Et bilde som inneholder tekst

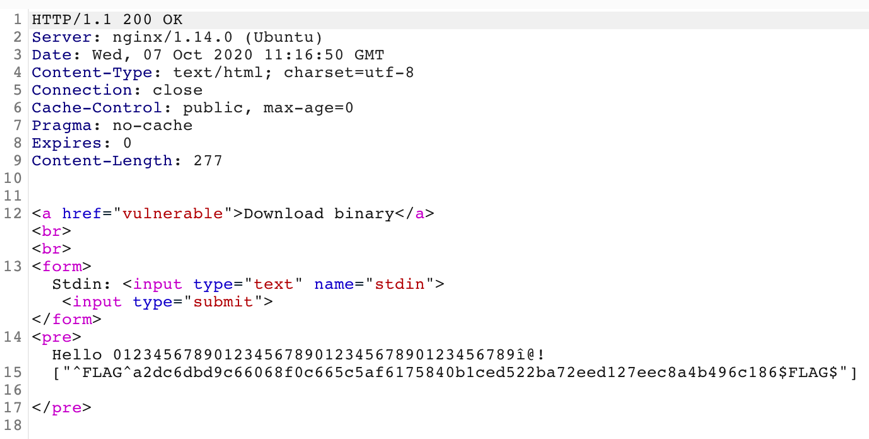
Automatisk generert beskrivelse

The address is therefor: ee064000.

In step 0 we found out that the file, vulnerable, is ELF 64bit LSB executable. Therefor we rewrite the address to 64 bit format: %ee%06%40%00%00%00%00%00

**Step 4:** Sent 40 chars + the flags memory address as input. The secret flag apers as the output.

Input: “0123456789012345678901234567890123456789%ee%06%40%00%00%00”

Output:

**Submitted the flag**: 