6 Security (fms27)

As an Elite Pentester, you have been engaged to attack a C program that Megacorp runs on an obsolete OS on a 32-bit x86 box, in the mistaken belief that their intrusion detection system offers sufficient protection. The target program accepts textual input from unauthenticated remote users. From prior intelligence you know that the input routine uses a word-aligned 64-byte buffer, that it is vulnerable to buffer overflow, and that modern buffer overflow countermeasures are not enabled. You can replicate the hardware and OS of the target in your lab, but you do not have a copy of the software. So as not to trigger the intrusion detection system, you may only send *one* attack input to the remote server. You have estimated that the buffer will appear at 0xfffe8200 plus or minus 4096 bytes, both endpoints included, and that the return address will be between 64 and 1024 bytes from the start of the buffer, both endpoints included. You wish to craft an attack input (as a sequence of bytes) that will result in the execution of a specific machine code payload (supplied, of length 113 bytes) on the remote machine. Your attack input consists of n_{ret} repetitions of your rewritten return address r, starting at offset $o_{ret} = 0$; followed by n_{nop} repetitions of the nop instruction, starting at offset $o_{nop} = 4 \cdot n_{ret}$; followed by your payload, starting at offset $o_{pl} = 4 \cdot n_{ret} + n_{nop}$; for a total file length $l = 4 \cdot n_{ret} + n_{nop} + 113$ bytes.

[*Note:* Correct numerical answers are important for this question. Please highlight your final numerical answers to distinguish them clearly from your scratch work and intermediate results.]

(*a*) Describe precisely all the absolute memory locations (hex values) where the return address of the vulnerable routine might appear and say how many there are (decimal value). [5 marks]

(*b*) Explain in detail how to compute n_{ret}, and do so (decimal value). [5 marks]

(*c*) Explain in detail how to compute n_{nop} and do so (decimal value), clarifying also why we need those nops in the first place. [5 marks]

(*d*) Explain in detail how to compute r, and do so (hex value). [5 marks]