Lab: Introductory Keygen

Environment Needed:

- Linux Virtual Machine (Recommend Ubuntu)

Links for any tools are available in the github under 'Tools'

1. In a terminal, navigate to ~/<where you downloaded the lab files>
2. We need to setup our keygenme application to be executable, run the command *chmod +x keygenme* to make it executable.
3. A common first step to reverse engineering – run the file and see what happens. This isn't always a good idea, but it is safe here. Run the file with the command "./*keygenme*".
4. Attempt to enter a name and a key.
5. Use objdump to examine the contents of *keygenme*.
6. Locate the key check function.
7. Look for calls to fgets and scanf – where are the username and key being stored on the stack?
8. Look for calls to printf – at what point is the key accepted?
9. Manually trace accesses to the username and key. It may help to copy the output of objdump into a text editor so that you can make notes as you go. At what points is an incorrect key rejected?
10. By watching where values from the key and name are used, decipher the checks on the key. You may need to look up some new x86 registers and instructions (http://ref.x86asm.net/coder32.html AND https://www.felixcloutier.com/x86/index.html). With hundreds (by some counts, thousands) of instructions in x86, even the best crackers will have to look up instructions once in a while.

Task 1:

Generate a valid key for your name. This can be done with a programming language of your choosing, but can also be solved by hand.

Task 2 (Bonus):

Old school crackers commonly released keys that contained their name, l337 text, or interesting magic numbers. This would mark the key as their own, and show that they had not only cracked the program, but had sufficient control of the key to manipulate it themselves. In the spirit of cracking, find a username/key combination such that the first 4 numbers of the key are "1337".

X86-SOFTWARE-REVERSE-ENGINEERING-CRACKING-AND-COUNTER-MEASURES
ISBN: **1394199880**