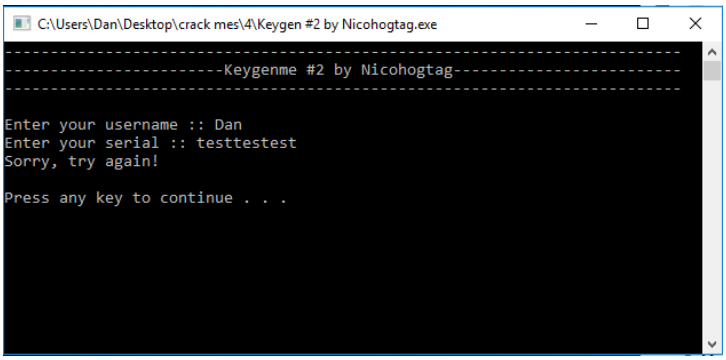
Final Turn In (extended): Crack Me Daniel Hoynoski

Friday, May 4th, 2018 **"Keygen #2 by Nicohogtag.exe"** Net ID: hoynosk2

1. I start off this crack-me yet again with just playing around with it. I opened it up and sought to see what the task was:



2. Another serial based crack-mes. In the last Crack-Me, I chose to self-keygen (i.e. find just one solution), but for this Crack-Me, I will develop a program to generate keys for me.
3. Opening this up in OllyDBG, I find the area of immediate interest after searching for reference strings:

```
.  C74424 04 ED( MOV DWORD PTR [ESP+4],Keygen_#.004400ED  ASCII "Enter your username :: "
.  C70424 C0334( MOV DWORD PTR [ESP],Keygen_#.004433C0
.  E8 81AD0300   CALL Keygen_#.0043C1B8
.  8D45 F8       LEA EAX,DWORD PTR [EBP-8]
.  894424 04     MOV DWORD PTR [ESP+4],EAX
.  C70424 60344( MOV DWORD PTR [ESP],Keygen_#.00443460
.  E8 3EBB0300   CALL Keygen_#.0043CF88
.  C74424 04 05( MOV DWORD PTR [ESP+4],Keygen_#.00440105  ASCII "Enter your serial :: "
.  C70424 C0334( MOV DWORD PTR [ESP],Keygen_#.004433C0
.  E8 5AAD0300   CALL Keygen_#.0043C1B8
.  8D45 F4       LEA EAX,DWORD PTR [EBP-C]
.  894424 04     MOV DWORD PTR [ESP+4],EAX
.  C70424 60344( MOV DWORD PTR [ESP],Keygen_#.00443460
.  E8 0F6E0200   CALL Keygen_#.00428280
.  C745 E0 0000( MOV DWORD PTR [EBP-20],0
>  837D E0 09    CMP DWORD PTR [EBP-20],9
.v 77 26         JA SHORT Keygen_#.004014A4
.  89E8          MOV EAX,EBP
.  0345 E0       ADD EAX,DWORD PTR [EBP-20]
.  83E8 08       SUB EAX,8
.  0FB600        MOVZX EAX,BYTE PTR [EAX]
.  8845 EF       MOV BYTE PTR [EBP-11],AL
.  0FBE55 EF     MOVSX EDX,BYTE PTR [EBP-11]
.  8D45 F0       LEA EAX,DWORD PTR [EBP-10]
.  0110          ADD DWORD PTR [EAX],EDX
.  8B55 F0       MOV EDX,DWORD PTR [EBP-10]
.  8D45 E8       LEA EAX,DWORD PTR [EBP-18]
.  0110          ADD DWORD PTR [EAX],EDX
.  8D45 E0       LEA EAX,DWORD PTR [EBP-20]
.  FF00          INC DWORD PTR [EAX]
.^ EB D4         JMP SHORT Keygen_#.00401478
>  8B45 F0       MOV EAX,DWORD PTR [EBP-10]
.  8B55 E8       MOV EDX,DWORD PTR [EBP-18]
.  01C2          ADD EDX,EAX
.  8B45 E4       MOV EAX,DWORD PTR [EBP-1C]
.  0FAFC2        IMUL EAX,EDX
.  8945 E4       MOV DWORD PTR [EBP-1C],EAX
.  8B55 F0       MOV EDX,DWORD PTR [EBP-10]
.  8B45 E4       MOV EAX,DWORD PTR [EBP-1C]
.  89C1          MOV ECX,EAX
.  29D1          SUB ECX,EDX
.  8B55 E8       MOV EDX,DWORD PTR [EBP-18]
.  89D0          MOV EAX,EDX
.  C1F8 1F       SAR EAX,1F
.  C1E8 1F       SHR EAX,1F
.  8D0402        LEA EAX,DWORD PTR [EDX+EAX]
.  89C2          MOV EDX,EAX
.  D1FA          SAR EDX,1
.  89D0          MOV EAX,EDX
.  01C0          ADD EAX,EAX
.  01D0          ADD EAX,EDX
.  C1E0 02       SHL EAX,2
.  01D0          ADD EAX,EDX
.  8D1401        LEA EDX,DWORD PTR [ECX+EAX]
.  8B45 E4       MOV EAX,DWORD PTR [EBP-1C]
.  0FAFC2        IMUL EAX,EDX
.  8945 E4       MOV DWORD PTR [EBP-1C],EAX
.  837D E4 00    CMP DWORD PTR [EBP-1C],0
.v 79 0F         JNS SHORT Keygen_#.004014FD
.  8B45 E4       MOV EAX,DWORD PTR [EBP-1C]
.  BA 00000000   MOV EDX,0
.  29C2          SUB EDX,EAX
.  89D0          MOV EAX,EDX
.  8945 E4       MOV DWORD PTR [EBP-1C],EAX
>  8B45 F4       MOV EAX,DWORD PTR [EBP-C]
.  3B45 E4       CMP EAX,DWORD PTR [EBP-1C]
.v 75 16         JNZ SHORT Keygen_#.0040151B
.  C74424 04 1C( MOV DWORD PTR [ESP+4],Keygen_#.0044011C  ASCII "Congrats, now write me keygen!"
```

4. With the above x86 code and a pencil and paper (left out OllyDBG comments this time, because it was faster), I mapped out the math behind using the name to generate the serial. The code is available on Github, but here is a screenshot:

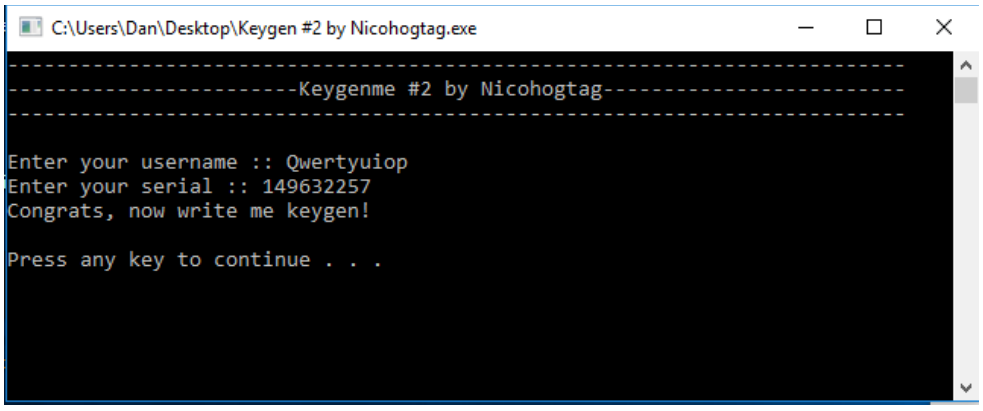
```
int main() {
   char * name = "Qwertyuiop"; // Must be length 10 to work b/c loop below
   int serialNumber = 0;
   int sideNumber = 526489; //Obtains from crack-me binary in OllyDBG

   for(int i = 0; i < 10; i++){
       serialNumber += (unsigned char) name[i];
       sideNumber += serialNumber;
   }

   int sum = serialNumber + sideNumber;
   int halfSide = sideNumber / 2;
   serialNumber = ((sum << 2) + (sum << 1) + sum) - serialNumber + (halfSide << 3) + (halfSide << 2) + halfSide;
   serialNumber *= ((sum << 2) + (sum << 1) + sum);

   if(serialNumber < 0){
       serialNumber -= serialNumber;
   }
   printf("%d", serialNumber);
}
```

5. For the username "Qwertyuiop" above, it generates a corresponding serial of "149632257" and allows me to finish this Crack-Me:



Finished.

Final Remarks:

I always get a thrill with finding the solution to these reverse engineering problems. Especially with the solution above, which can be used to generate any solution for any username (above 10 characters (see screenshot for details on this)), it was fun finding all the solutions and not just one like

before. Even if this took me some time to decode (i.e. large chunks of assembly can sometimes take me hours to read), I still enjoyed it.