演示用 QuickView 修改 password.exe:

QuickView 工具只能在 xp 及 dosbox86 下工作；hiew(http://10.71.45.100/bhh/hiew.zip)可以在 Win7/8/10 下工作，但功能不如 QuickView。

①下载 dosbox86 虚拟机:

http://10.71.45.100/bhh/DosBox86.rar

解压缩后，生成 DosBox86 文件夹；

②双击 dosbox86\dosbox86.exe 会自动运行 16 位汇编语言的集成环境，选菜单 file->exit 退出集成环境并回到虚拟 dos 系统；

此时在命令行中会显示提示符： c:\masm

注意此文件夹是从 dosbox86\masm 虚拟出来的。

该文件夹里包含了 16 位汇编语言相关的所有编译调试工具，同时还包括 quickview 工具。

③把 password.exe 拷到 dosbox86\masm 中

④在虚拟 dos 系统下，输入以下命令:

qv password.exe

⑤敲回车切换到汇编模式下，再按 F2 可以在 16 汇编及 32 汇编之间切换。

⑥按 F7 搜索机器码： 83 C4 08 85 C0 75 0F，搜到后把 75 0F 改成 90 90；也可以先暂时不改 75 0F,而是按 Tab 键跳到右侧汇编指令那里，再输入

nop

nop

要是不小心改坏了，可以按 Tab 键跳回左侧机器语言那里，再按 F3 撤销修改。

⑦按 Alt+F9 可以保存修改。

⑧输入以下命令关闭 dosbox:

exit

regtest 破解

http://10.71.45.100/bhh/regtest.rar

regtest.rar 解压缩后，里面有一个 reg.exe，

用 od 打开它进行调试:

在 windows 平台，用 C 或 C++编程时会调用两类函数:

(1) 库函数: gets puts scanf printf strcmp

这些函数经过编译后，函数名会消失，变成函数的地址；

(2) Windows 系统内核的函数: MessageBoxA, GetCommandLineA, LoadLibrary, GetProcAddress

象 MessageBox 这种跟字符串相关的函数通常有 2 个版本:

①ansi: MessageBoxA
字符串"abc"用ansi格式表示为:0x61,0x62,0x63,0x00
②unicode: MessageBoxW
字符串"abc"用unicode格式表示为:
0x61,0x00, 0x62,0x00, 0x63,0x00, 0x00,0x00
系统内核的函数又叫做api(application program interface)。
编译的时候，源代码中的MessageBox会被替换成MessageBoxA。也就是说，在源代码中要调用ansi版本的函数，既可以写成MessageBox也可以写成MessageBoxA。

当某个exe里面调用了这些函数时,这些函数的函数体并不会编译进入exe里面，而是独立存在于操作系统内核中，并且它们的地址在系统启动后是固定的。
在od的代码窗中按ctrl+g并输入MessageBoxA就可以定位到该函数的首地址处，此处按F2设一个断点。
接下去点run按钮让reg.exe运行。
输入注册码1234点确定后，会断在刚才所设断点上，
此时可以观察到寄存器eip刚好等于断点地址，因为eip
表示当前将要执行的指令的地址。
一直按F8单步走到retn处(有些win10的电脑可能会在到达retn前调用MessageBoxExA处卡住,原因是看不到那个弹框或点不掉那个弹框,此时建议不要F8步过而是在retn处再设一个断点,再点run按钮运行)，再按F8就会回到调用者那里。
现在来到此处:

```
0040C85A   pop     esi
0040C85B   retn    0xC
```

按两次F8把函数40C82C走完
结果来到此处:

```
004013F0  mov     ecx, [ebp-0xC]
```

按PgUp可以看到以下可疑指令:

```
004013D1  mov     eax, [esi+0x60]
004013D4  mov     ecx, [esi+0x5C]
004013D7  xor     eax, 0x1999AA98
004013DC  cmp     eax, ecx
004013DE  je      short 00401401
```

注意C语言的异或运算^在汇编语言里是: xor
接下去在地址4013D1处设断点，F8单步两次并观察

eax 及 ecx 的值，可以发现 eax 是我们乱输的注册码 1234，而 ecx 就是信息码。根据上述程序片断，可以整理出以下结论：

(sn ^ 0x1999AA98) == 特征码 ➔

sn = 特征码 ^ 0x1999AA98 = 1005708783