Crackme: synamics's Xrockmr
Difficulty: Easy
Tools of trade: gdb, Hex Editor, objdump

- This crackme was quite simple and obvious. Solving can be divided in two phases. The first phase is an anti-debugging trick and second is the password comparison phase.
- These two phases are apparently spread across the three methods viz. `_init_, main, _finit_`
- The anti-debugging trick has been placed in `_init_` phase. Lets examine it here:

```
8048584:    55                        push   %ebp
8048585:    89 e5                     mov  %esp,%ebp
8048587:    83 ec 18                  sub  $0x18,%esp
804858a:    c7 04 24 cc 88 04 08     movl   $0x80488cc,(%esp)
8048591:    e8 9a fe ff ff            call   8048430 <system@plt>
                                     "killall -q gdb"
8048596:    c7 04 24 db 88 04 08     movl   $0x80488db,(%esp)
804859d:    e8 8e fe ff ff            call   8048430 <system@plt>
                                     "killall -q strace"
```

- We can observe in the above mentioned piece of code, the specified strings are being loaded and a call is being made to `_system` function. Both of these commands kill all the instances of any debugger.
- Hence in order to debug it further, we got to edit it in hex editor and change the commands (strings) to something similar without changing their start/end positions. I chose some stupid strings like `"grep hell well", "grep hell heavens".` (Funny, ain't they?)
- After this step, we can successfully step through the `_init_` portion and break into `main` to this line here:

```
80487a3:    e8 78 fc ff ff            call   8048420 <puts@plt>
                                     "What is the password?"
80487a8:    a1 30 a0 04 08            mov  0x804a030,%eax
80487ad:    89 44 24 08               mov  %eax,0x8(%esp)
80487b1:    c7 44 24 04 08 00 00     movl   $0x8,0x4(%esp)
80487b8:    00
80487b9:    c7 04 24 3c a0 04 08     movl   $0x804a03c,(%esp)
80487c0:    e8 2b fc ff ff            call   80483f0 <fgets@plt>
80487c5:    0f b6 05 40 a0 04 08     movzbl 0x804a040,%eax
80487cc:    3c 6f                     cmp  $0x6f,%al
80487ce:    74 07                     je   80487d7 <main+0x1ba>
80487d0:    c6 05 3d a0 04 08 61     movb   $0x61,0x804a03d
80487d7:    b8 00 00 00 00            mov  $0x0,%eax
80487dc:    c9                        leave
80487dd:    c3                        ret
```

- We above piece of code, outputs "`What is the password?`" and then takes in input from `stdin` via `_fgets` function. After that on stepping further, the fifth character of the password is compared to '`o`'. If they're equal then the second character of the password is replaced with '`a`' else it proceeds to execute the `_finit_` portion.
- Lets break the `_finit_` portion here:

```
8048544:    55                       push   %ebp
8048545:    89 e5                    mov  %esp,%ebp
8048547:    83 ec 18                 sub  $0x18,%esp
804854a:    0f b6 05 3c a0 04 08    movzbl 0x804a03c,%eax
8048551:    3c 68                    cmp  $0x68,%al
8048553:    75 2d                    jne  8048582 <_finit_+0x3e>
8048555:    0f b6 05 3d a0 04 08    movzbl 0x804a03d,%eax
804855c:    3c 65                    cmp  $0x65,%al
804855e:    75 22                    jne  8048582 <_finit_+0x3e>
8048560:    0f b6 05 3e a0 04 08    movzbl 0x804a03e,%eax
8048567:    3c 6c                    cmp  $0x6c,%al
8048569:    75 17                    jne  8048582 <_finit_+0x3e>
804856b:    0f b6 05 3f a0 04 08    movzbl 0x804a03f,%eax
8048572:    3c 6c                    cmp  $0x6c,%al
8048574:    75 0c                    jne  8048582 <_finit_+0x3e>
8048576:    c7 04 24 b0 88 04 08    movl   $0x80488b0,(%esp)
804857d:    e8 9e fe ff ff           call   8048420 <puts@plt>
8048582:    c9                       leave
8048583:    c3                       ret
```

- We can see the that our password string is being compared to the string "`hell`" character by character. Hence in order to avoid jumping off the course from here, we have to ensure that the fifth character matches '`o`' so that the second character remains unchanged (We can observe that piece of code at the end of `main`).
- And we can finally conclude that the password is "`hello`"`<any string here>`. i.e Any string of 0 or more in length placed after "`hello`" and that will satisfy the crackme.

End of tutorial,
Cheers, **whizz**.