Cracking With +Vip-Vop

Using W32dasm Softice to crack Password Trecker Deluxe 3.45

(Program can be fround at http://www.bigfoot.com/~ptd)

At first I tried the same way I had done with Pspli97, but as you said in your mail it was pretty confusing. There seemed to be multiple flags etc. Well I screwed with it that way for about 1 hour, then decided to go to sleep and try again later, because sometimes you need to take a break from a program so you can try a different approach the next time. So when I came back to PasswordTracker, I tried a different approach. I clicked on "Help", then "Registration Number". I entered "+VipVop" and "666-555" for the number, then clicked "Ok". A message box pops up saying "Invalid registration number". >From there I had two options, either do a "dead-list" approach and open up the program in W32dsm and search for the string "Invalid regist...." or try in backtrack in softice by doing a "bpx messageboxa" then pressing F12 a couple of times after the box popped up. Wrongly, I tried the soft-ice approach first. After messing with it that way for a few minutes I realized I wasn't getting anywhere so I decided to try and search for the message box string in W32dsm. So after searching for the string "Invalid registration number" this is what I found:

```
:004197FA 8B542404              mov edx, dword ptr [esp+04]
:004197FE 8B442408              mov eax, dword ptr [esp+0   8]
:00419802 52                    push edx
:00419803 50                    push eax
:00419804 E8C7050000            call 00419DD0
:00419809 83C408                add esp, 00000008
:0041980C 83F82D                cmp eax, 0000002D
:0041980F 7509                  jne 0041981A
:00419811 8BCE                  mov ecx, esi
:00419813 E8E3770100            call 00430FFB
:00419818 EB21                  jmp 0041983B

* Referenced by a (U)nconditional or (C)onditional Jump at    Address:
|:0041980F(C)
|
:0041981A 6A00                  push 00000000
:0041981C 6A30                  push 00000030

* Possible StringData Ref from Data Obj ->"Invalid registra   tion number."
                                |
:0041981E 6894B34600            push 0046B394
:00419823 E847330200            call 0043CB6F
```

Well right away we see that its referenced by a conditional jump (the (C)), so we look at where the jump came from (0041980F). So look at the following lines:

```
:0041980C 83F82D                cmp eax, 0000002D
:0041980F 7509                  jne 0041981A
```

All it does is copmare eax to 2d (hex), and if it isnt equal it jumps to the "Invalid registration number." text. Well how does eax get set to whatever its set to? Scrolling up above the cmp a few lines we see the following call:

```
:004197FA 8B542404              mov edx, dword ptr [esp+04]
:004197FE 8B442408              mov eax, dword ptr [esp+08]
:00419802 52                    push edx
:00419803 50                    push eax
:00419804 E8C7050000            call 00419DD0
```

Since thats the only call around its likely that that call sets eax to whatever. Now just by looking at the code, we can tell its sending 2 different paramters to a call (the push edx and push eax). Well since we are in the registration scheme, what do you want to bet those pushs are pushing our name and reg number to the call? You can do a "bpx 00419802" in s-ice, then "d edx" and "d eax" to prove that yes, that is our name and reg number.

So reviewing what we know now, whatever code is called at 00419DD0 accepts our reg name and reg number, and if they are correct makes eax equal 2d and returns. Now its also likely that most programmers wouldnt write the same function twice to do the same thing, so chances are if we patch the code at 00419dd0 the whole program will be registered. So lets try that. Clear all your breakpoints in S-Ice, do a "bpx 00419dd0" and enter whatever you want for your name and reg number. Sice will pop up and you will see this:

```
* Referenced by a CALL at Addresses:
|:00419804   , :00419D2F
|
:00419DD0 6AFF                  push FFFFFFFF
:00419DD2 68D8234500            push 004523D8
:00419DD7 64A100000000          mov eax, dword ptr fs:[0000000]
:00419DDD 50                    push eax
:00419DDE 648925000000000       mov dword ptr fs:[0000000], esp
:00419DE5 83EC14                sub esp, 00000014
:00419DE8 8B442424              mov eax, dword ptr [esp+24]
:00419DEC 56                    push esi
:00419DED 50                    push eax
:00419DEE 8D4C2408              lea ecx, dword ptr [esp+08]
:00419DF2 E8C9FAFFFF            call 004198C0
:00419DF7 8B54242C              mov edx, dword ptr [esp+2C]
:00419DFB 51                    push ecx
:00419DFC 8BCC                  mov ecx, esp
:00419DFE 8964242C              mov dword ptr [esp+2C], esp
:00419E02 52                    push edx
:00419E03 C744242800000000      mov [esp+28], 00000000
:00419E0B E8EAB00100            call 00434EFA
```

CONTINUED ON NEXT PAGE:

```
:00419E10 8D4C2408              lea ecx, dword ptr [esp+08]
:00419E14 E807FCFFFF            call 00419A20
:00419E19 8D4C2404              lea ecx, dword ptr [esp+04]
:00419E1D 8BF0                  mov esi, eax
:00419E1F C7442420FFFFFFFF      mov [esp+20], FFFFFFFF
:00419E27 E8A4FBFFFF            call 004199D0
:00419E2C 8B4C2418              mov ecx, dword ptr [esp+18]
:00419E30 8BC6                  mov eax, esi
:00419E32 64890D00000000        mov dword ptr fs:[00000000], ecx
:00419E39 5E                    pop esi
:00419E3A 83C420                add esp, 00000020
:00419E3D C3                    ret
```

Notice you cant see eax being set to anything anywhere, or at least being set to 2d (possibly set anyway, if our reg code is right),and we are too lazy to trace through all this code, so lets do it the easy way. in s-ice type "a 00419dd0", which lets us change the instructions at 00419dd0. Well all we want to do is make eax equal to 2d, then return from the call, so type:

```
mov eax,2d
ret
```

And now our call is patched. Notice it now says thank you for registering, and when you click on the about screen it says registered to (your name). So open up PwTrkr.exe in your favorite hex editor, and search and replace the following bytes:

SEARCH: 6AFF68D82334500 (the first 2 instructions of 00419dd0)
REPLACE: B82D000000CE90 (the mov eax,2d, the ret, and a NOP to keep it the right length)

One last thing, if you just crack the exe but dont have anything in your registry for the reg name and number, it will stay say unregistered. So make sure to enter anything you want for the "Help" "Registration Number" part.--====================987654321_0==_
Content-Type: text/plain; charset="us-ascii"

Written by: +Vip-Vop
Published by: Splatter Industries,Inc.