

C# Protect the Password inside a TextBox

May 4, 2013 / security



First of all, the **TextBox** Control is not a secure way to hold passwords - so don't use it as it is by default. Even with a **PasswordChar**, the real password is still visible if you use a tool like **Spy++**.

The Flaw (Explained)

As an example, take a simple **TextBox** with **PasswordChar = '*'**.
Now, if you use **Spy++** and inspect this application, you can get the password.

Newsletter

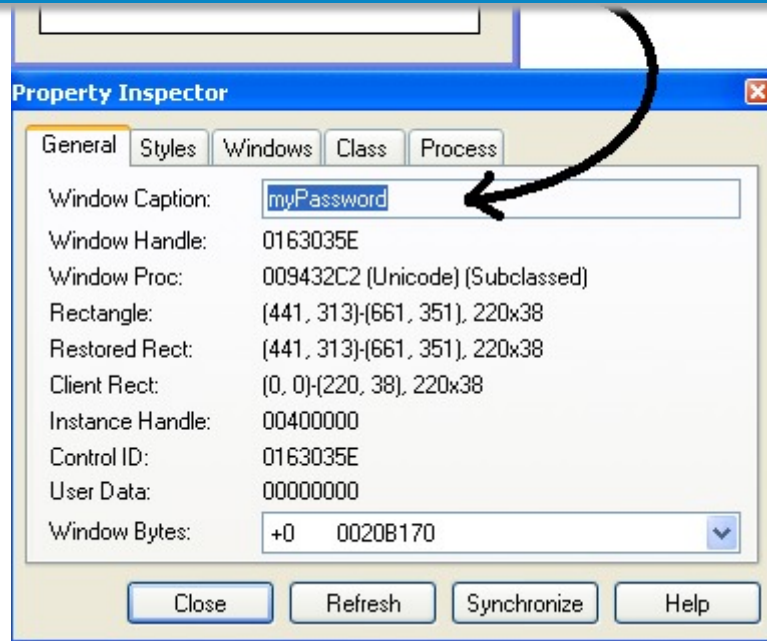


I know I don't publish very often - but you can find out when that happens:

[coding.vision newsletter](#)

Categories

ai files interface
miscellaneous networking
numerical-methods
personal security
tips-and-tricks



Recovering values from Password Textboxes with Spy++

)

But...why is this happening?

Spy++ sends the message **WM_GETTEXT (0x000D)** to the TextBox, and the control just returns its value. It's actually working as intended and it reacts just like any other control, that obeys any message received.

Fixing this issue...

If the Text property is called, it will send an WM_GETTEXT message, so it will surely be an internal (safe) call. But if that message is received and the Text property wasn't called, then it might be risky to return the password, so we'll not process that message.

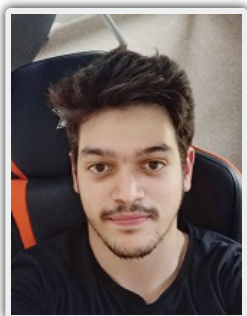


I wrote a "safer" **TextBox** here, just to show you the idea, feel free to write your own or simply improve this one.



```
1  class ProtectedTextBox : TextBox
2  {
3      // the malicious message, that needs to be handled
4      private const int WM_GETTEXT = 0x000D;
5
6      // 'true' if the messages are sent from our program (from T
7      // 'false' if they're sent by anything else
8      bool allowAccess { get; set; }
9
10     public override string Text // overriding Text property
11     {
12         get
13         {
14             allowAccess = true; // allow WM_GETTEXT (because
15             return base.Text; //this sends the message above i
16         }
17         set
18         {
19             base.Text = value;
20         }
21     }
```

```
25         if (m.Msg == WM_GETTEXT) // if the message is WM_GETTE
26         {
27             if (allowAccess) // and it comes from the Text proc
28             {
29                 allowAccess = false; //we temporary remove th
30                 base.WndProc(ref m); //and finally, process th
31             }
32         }
33         else
34             base.WndProc(ref m);
35     }
36 }
```



Dan Sporici

Junior Security Researcher @PRECIS (UPB), Master's Student in Advanced Cybersecurity and Graduate Teaching Assistant.

coding.vision

