

معمولا وقتی صحبت از هک می‌شود بیشتر وب سایتها مدنظر هستند و کمتر به نرم افزارهای محیط desktop توجه می‌شود اما در محیط desktop هم امنیت بسیار مهم است.

یکی از مواردی که در رابطه با windows form ها (دات نت و غیر دات نت) بسیار به آن برخورد کرده ام اعمال دسترسی کاربر به قسمتهای مختلف فرم با مخفی یا غیر فعال کردن دکمه‌ها و کنترل‌های فرم است. به این صورت که در هنگام لود فرم اصلی با توجه به دسترسی کاربر بعضی منوها مخفی می‌شوند و در لود هر فرم هم بعضی از دکمه‌ها و کنترل‌های روی فرم مخفی یا غیر فعال می‌شوند. همین!

اما اگر کاربر به کمک ابزاری بتواند منوها و کنترل‌های مخفی فرم را نمایش دهد و منوها و کنترل‌های غیرفعال را فعال کند چی؟

کلاس زیر را در نظر بگیرید :

```
public class HackWindowsForms
{
    struct POINTAPI
    {
        public int X;
        public int Y;
    }

    [DllImport("user32.dll")]
    static extern int EnumChildWindows(int hWndParent, EnumChildCallbackDelegate enumChildCallback,
int lParam);
    [DllImport("user32.dll")]
    static extern int EnableWindow(int hwnd, int fEnable);
    [DllImport("user32.dll")]
    static extern int WindowFromPoint(int x, int y);
    [DllImport("user32.dll")]
    static unsafe extern int GetCursorPos(POINTAPI* lpPoint);

    delegate int EnumChildCallbackDelegate(int hwnd, int lParam);

    public static void EnableThisWindowControls()
    {
        unsafe
        {
            POINTAPI cursorPosition = new POINTAPI();
            GetCursorPos(&cursorPosition);
            int winWnd = WindowFromPoint(cursorPosition.X, cursorPosition.Y);
            EnumChildWindows(winWnd, EnumChildCallback, 0);
        }
    }

    static int EnumChildCallback(int hwnd, int lParam)
    {
        EnableWindow(hwnd, 1);
        EnumChildWindows(hwnd, EnumChildCallback, 0);
        return 1;
    }
}
```

یک پروژه ویندوز فرم ایجاد کنید و دکمه ای روی فرم قرار دهید. برای MouseUp آن کد زیر را بنویسید :

```
private void button1_MouseUp(object sender, MouseEventArgs e)
{
    HackWindowsForms.EnableThisWindowControls();
    MessageBox.Show("را فشار دهید space حالا روی دکمه مورد نظر کلیک کنید و کلید", "",
    MessageBoxButtons.OK, MessageBoxIcon.Asterisk, MessageBoxDefaultButton.Button1,
    MessageBoxOptions.RtlReading);
}
```

```
}
```

حالا اگر برنامه را اجرا کنید و روی دکمه کلیک کنید و ماوس را نگهدارید سپس روی نوار عنوان پنجره مورد نظر رها کنید، تمام کنترل‌های روی آن پنجره فعال خواهند شد! نمونه کد مشابهی هم برای نمایش منوها و کنترل‌های مخفی شده می‌توان نوشت.

اما راه حل چیست؟

قبل از اجرای هر عملی که احتیاج به دسترسی دارد، دسترسی کاربر چک شود و فقط به غیر فعال و مخفی کردن کنترل‌ها بسنده نشود. مثلا اگر دکمه ویرایش را غیر فعال کرده ایم در کلیک آن هم دسترسی چک شود.

نظرات خوانندگان

نویسنده: Blackstar
تاریخ: ۱۳۹۱/۰۴/۲۵ ۰:۴۹

بهتر نیست اون منو ویا شئی رو dispose کرد

نویسنده: رحمت اله رضایی
تاریخ: ۱۳۹۱/۰۴/۲۵ ۱۰:۱۱

از مخفی و غیرفعال کردن بهتر است (البته تجربه این روش رو ندارم). اما در هر صورت راه حل اصلی همان است که گفتم.

نویسنده: صابر فتح اللهی
تاریخ: ۱۳۹۱/۰۴/۲۵ ۱۰:۳۷

مطمعنا بیشتر از 80 درصد نرم افزارهای داخلی اینگونه طراحی شدن حاضرم از نرم افزارهای اتوماسیون اداری شرکت‌های مهمی نام ببرم که به این روش کار کردن چون دیدم دارم می‌گم

روشی که آقای رضایی می‌گن درسته، من کمتر تحت ویندوز کار کردم اما همیشه روش آقای رضایی انجام دادم

در کل پست به جایی بود ممنون

نویسنده: naser faraji
تاریخ: ۱۳۹۱/۰۴/۲۵ ۱۷:۲۹

ممنون از مقاله زیباتون. بله حرف شما کاملا درسته و بسیاری از نرم افزارهای مالی و حسابداری ایرانی به همین روش نوشته شده!

نویسنده: Mohsen
تاریخ: ۱۳۹۱/۰۴/۳۰ ۱۸:۳۵

سلام

مشکل درست است ولی راه حل ساده است.

بهتر است بر اساس دسترسی کاربر مورد نظر منوها و کنترل‌های روی صفحه در هنگام اجرا ایجاد گردند.

در دات نت کنترل (FlowLayoutPanel) هم برای مرتب سازی خودکار کنترل‌های ایجاد شده در هنگام اجرا وجود دارد که قضیه را بسیار ساده می‌کند.