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

Tématická oblast: PowerShell, Alternate stream

Jméno a číslo studenta: Adam Šárek (SAR0083)
Datum vypracování: 08.10.2021

Zadání:

1) Seznamte se s mechanismy „streamů“ (především pak „alternate streamu“), které jsou součástí NTFS file systému.
Ve svém testovacím prostředí proveďte následující úlohy a dle zjištění odpovězte na otázky:

   a. Přes konzoli (cmd) proveďte zápis do alternativního streamu některého, Vámi zvoleného, souboru.
   Napište Vámi použitý příkaz:
   `type secret.txt > file.txt:secret.txt`

   b. Vypište data ze zapsaného alternativního streamu.
   Napište Vámi použitý příkaz:
   `more < file.txt:secret.txt`

   c. Podívejte se na normální a alternativní stream přes správce souborů Windows (explorer.exe) a preš vlastnosti souboru. **Jaké informace poskytuje o souboru zapsaném v alternativním streamu**:
   Neposkytuje žádné informace o alternativním streamu, pouze informace o souboru samotném.

   d. Ve výpisu adresáře přes konzoli zobrazte také alternativní streamy.
   Napište Vámi použitý příkaz:
   `dir /r`

   e. Co se stane s daty v alternativním streamu **při PŘESUNU přes správce souborů na jiné místo téhož diskového oddílu**?
   Zůstanou beze změny.

   f. Co se stane s daty v alternativním streamu **při KOPÍROVÁNÍ přes správce souborů na jiné místo téhož diskového oddílu**?
   Zůstanou beze změny.

   g. Co se stane s daty v alternativním streamu **při KOPÍROVÁNÍ na FAT32**?
   Tyto data zaniknou, protože FAT32 nepodporuje alternativní streamy.

   h. Vyzkoušejte zápis a čtení do alternativního streamu přes WinAPI. **Program odešlete společně s protokolem!**

2) Seznamte se se základy PowerShellu a s kódováním base64. Dále splňte následující úlohy a zodpovězte otázky:

a. Spusťte následující příkaz:

```
powershell.exe -EncodedCommand
VwByAGkAdABlAC0ASABvAHMAdAAgAC0ATwBiAGoAZQBjAHQAIAAiAEgAZQBsAGwAbw
AsACAAdwBvAHIAbABkACEAIgA7AA==
```

Co tento příkaz dělá?
Zakódovaný řetězec obsahuje příkaz:
Write-Host -Object "Hello, world!";

Tento příkaz vypíše „Hello, world!“.

b. Vytvořte vlastní powershell script a převeďte jej do base64. Činnost skriptu zvolte dle vlastního uvážení.
Svůj zakódovaný skript přiložte k protokolu.

c. Přidejte Váš skript do registrů jako parametr powershellu tak, aby se spouštěl při přihlášení uživatele (stačí skript přidat manuálně, nemusíte vytvářet program ke vkládání do registrů).
Udělejte screen z regedit.exe.

Závěr:

Diskutujte následující témata:

1) Jakým způsobem může malware využít alternativní streamy a proč jsou pro malware zajímavé?
2) Proč je powershell zajímavý pro tvůrce malware?
3) Proč je pro malware zajímé kódování base64?

Vypracování

1) Malware může alternativní streamy využít ke skrytí přidaných či nezbytných souborů pro svou vlastní činnost (např. keylog.txt v rámci keyloggeru). Případně může malware do alternativního streamu uložit škodlivou aplikaci. Tato aplikace nebude v systému snadno detekovatelná, jelikož bude nést název původního souboru a zároveň se tomuto souboru nijak nezmění jeho ukazovaná velikost. Pro běžného uživatele je tedy nemožné detekovat využití této vlastnosti souborového systému NTFS.

   Právě tím, jak skrytě lze pracovat s libovolnými soubory je důvod, proč je tato vlastnost pro malware velmi zajímavá.

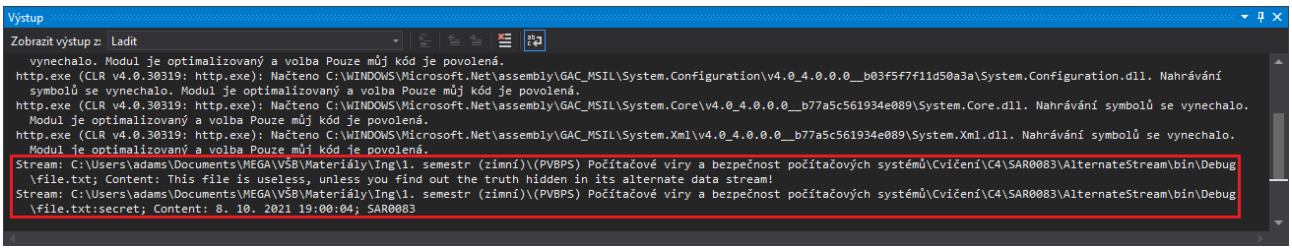
2) Powershell je pro tvůrce malware zajímavý tím, že v rámci jednoduché konzole poskytuje plnohodnotný přístup k určitým součástem systému Windows. Důležité je pro malware také možnost externího spouštění skriptů, což velmi zjednodušuje zneužití tohoto nástroje.

3) Pro malware je kódování base64 zajímavé tím, že jeho zakódovaný řetězec v sobě vizuálně skryje škodlivé příkazy, které uživatel nedokáže přečíst. V powershellu je však tento řetězec možné spustit díky parametru: `-EncodedCommand`.

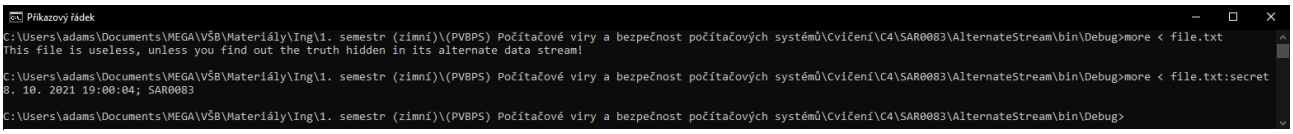
Snímky obrazovky

```
13      static class Program
14      {
15          private static DateTime date = DateTime.UtcNow;
16          private static string student = "SAR0083";
17          private static string fileName = Application.StartupPath + @"\file.txt";
18          private static string fileContent = "This file is useless, unless you find out the truth hidden in its alternate data stream!";
19          private static string fileAltStreamName = "secret";
20          private static string fileAltStreamContent = string.Join("; ", new string[] { date.ToString(), student });
21          private static string fileFullName = string.Join(":", new string[] { fileName, fileAltStreamName });
22
23          /// <summary>
24          /// Hlavní vstupní bod aplikace.
25          /// </summary>
26          [STAThread]
            Počet odkazů: 0
27          static void Main()
28          {
29              WriteFileStream(fileName, fileContent);
30              WriteFileStream(fileFullName, fileAltStreamContent);
31
32              DebugFileStream(fileName, ReadFileStream(fileName));
33              DebugFileStream(fileFullName, ReadFileStream(fileFullName));
34
35              Application.Run();
36          }
37
            Počet odkazů: 2
38          public static void DebugFileStream(string path, string content)
39          {
40              Debug.WriteLine("Stream: {0}; Content: {1}", path, content);
41          }
42
            Počet odkazů: 2
43          public static string ReadFileStream(string path)
44          {
45              using (StreamReader sr = new StreamReader(CreateFileStream(path, FileAccess.Read, FileMode.Open, FileShare.Read)))
46              {
47                  return sr.ReadToEnd();
48              }
49          }
50
            Počet odkazů: 2
51          public static void WriteFileStream(string path, string content)
52          {
53              using (StreamWriter sw = new StreamWriter(CreateFileStream(path, FileAccess.Write, FileMode.OpenOrCreate, FileShare.Delete)))
54              {
55                  sw.Write(content);
56              }
57          }
58
            Počet odkazů: 2
59          public static FileStream CreateFileStream(string path, FileAccess access, FileMode mode, FileShare share)
60          {
61              SafeFileHandle handle;
62              try
63              {
64                  handle = CreateFile(path, access, share, IntPtr.Zero, mode, 0, IntPtr.Zero);
65              } catch(IOException e)
66              {
67                  throw e;
68              }
69
70              return new FileStream(handle, access);
71          }
72
73          [DllImport("kernel32.dll")]
            Počet odkazů: 1
74          public static extern SafeFileHandle CreateFile(
75              string lpFileName,
76              FileAccess dwDesiredAccess,
77              FileShare dwShareMode,
78              IntPtr lpSecurityAttributes,
79              FileMode dwCreationDisposition,
80              uint dwFlagsAndAttributes,
81              IntPtr hTemplateFile
82          );
83      }
84  }
```

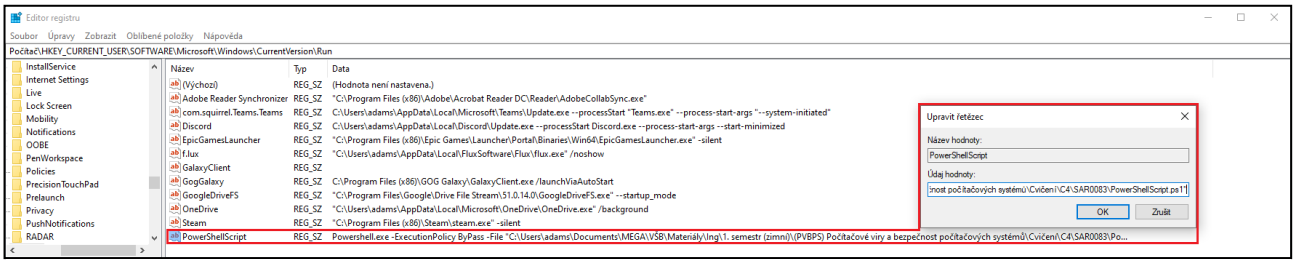
Obrázek 1 - Kód aplikace vytvořené pro příklad 1.h)



Obrázek 2 - Výpis konzole ve Visual Studiu po spuštění aplikace vytvořené pro příklad 1.h)



Obrázek 3 - Výpis konzole pro kontrolu souboru vytvořeného pro příklad 1.h)



Obrázek 4 – Snímek z regedit.exe s Powershell skriptem, který se spouští přes cmd.exe po startu Windows pro příklad 2.c)