Počítačové viry a bezpečnost počítačových systémů
Protokol z předmětu (3b)

Tématická oblast: Analýza malware – PE hlavičky a závislosti souborů

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

Jméno a číslo studenta: Daniel Trnka, TRN0038
Datum vypracování: 19. 4. 2019

Zadání:

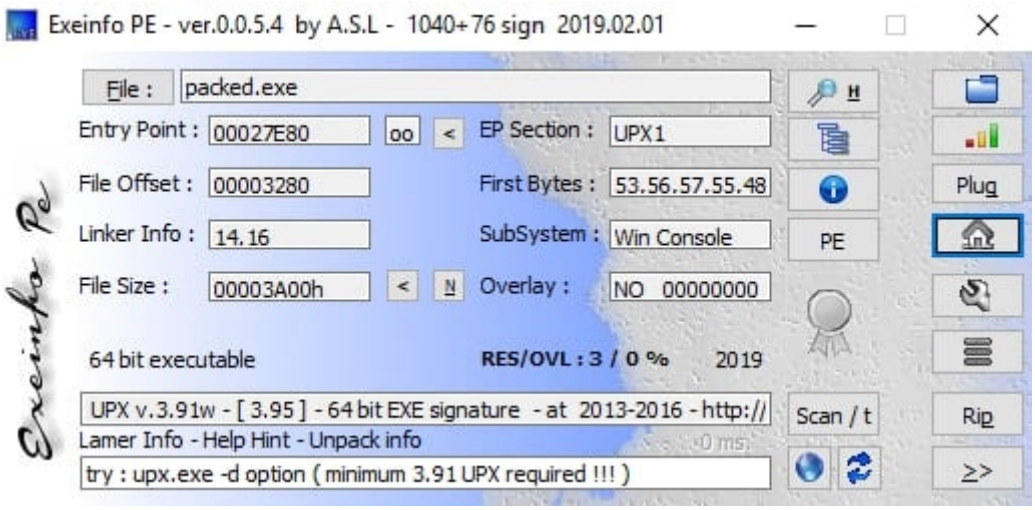
Každému studentovi bude na cvičení přiřazen vzorek malware.
Protokol vypracujte pouze pro Váš malware.
Malware: Sample5

POZOR! V rámci cvičení budete pracovat se skutečnými vzorky malware. Veškerou práci provádějte pouze v poskytnutém virtuálním prostředí.
Šíření malware je zakázáno!

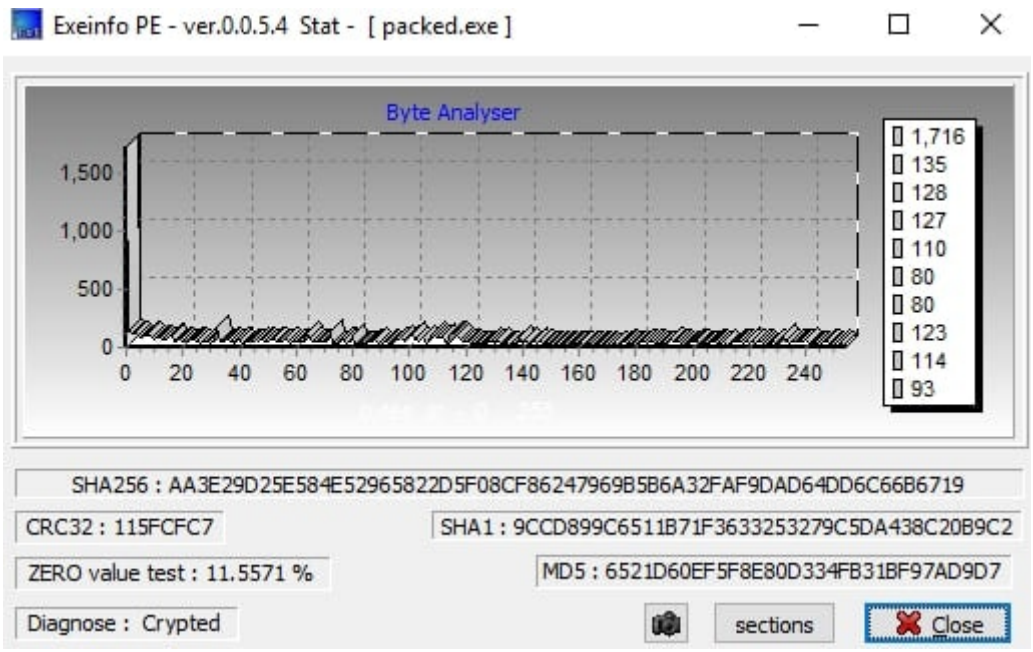
1) Nastudujte různé techniky, jež se užívají ke skrytí těla malware. Především pak techniky typu „obfuscation“ a „packing“. Stáhněte si libovolný packer (takový, který slouží ke komprimaci či šifrování spustitelných souborů) a zabalte Váš keylogger, který jste v rámci předchozích cvičení vytvořili. **Odpovězte na otázky:**

   - Jaký packer jste použili?
     Upx, protože se jedná o vyvíjený open source packer. Nepodporuje ale „packing“ C# aplikací, proto byl v rámci práce zabalen zjednodušený KeyLogger naimplementovaný v C.

   - Stáhněte program ExeInfo (www.exeinfo.xn.pl). Jak detekoval tento program Vámi užitou kompresní metodu? PŘILOŽTE SCREENSHOT.
     Program Execinfo detekoval packer upx správně.
     Sign. 4400 : [x64 - UPX exe - NRV2E/7 compression]



Frekvence jednotlivých bajtů je z následujícího grafu vcelku rovnoměrná, což může značit, že je soubor zašifrován.



- Na jaké další kompresní metody (typy packerů) spustitelných souborů jste během studia narazili? **Vypište alespoň 3 další:**
  Packer může spustitelný soubor komprimovat nebo navíc i šifrovat. Dále může například být několik packerů vnořených.

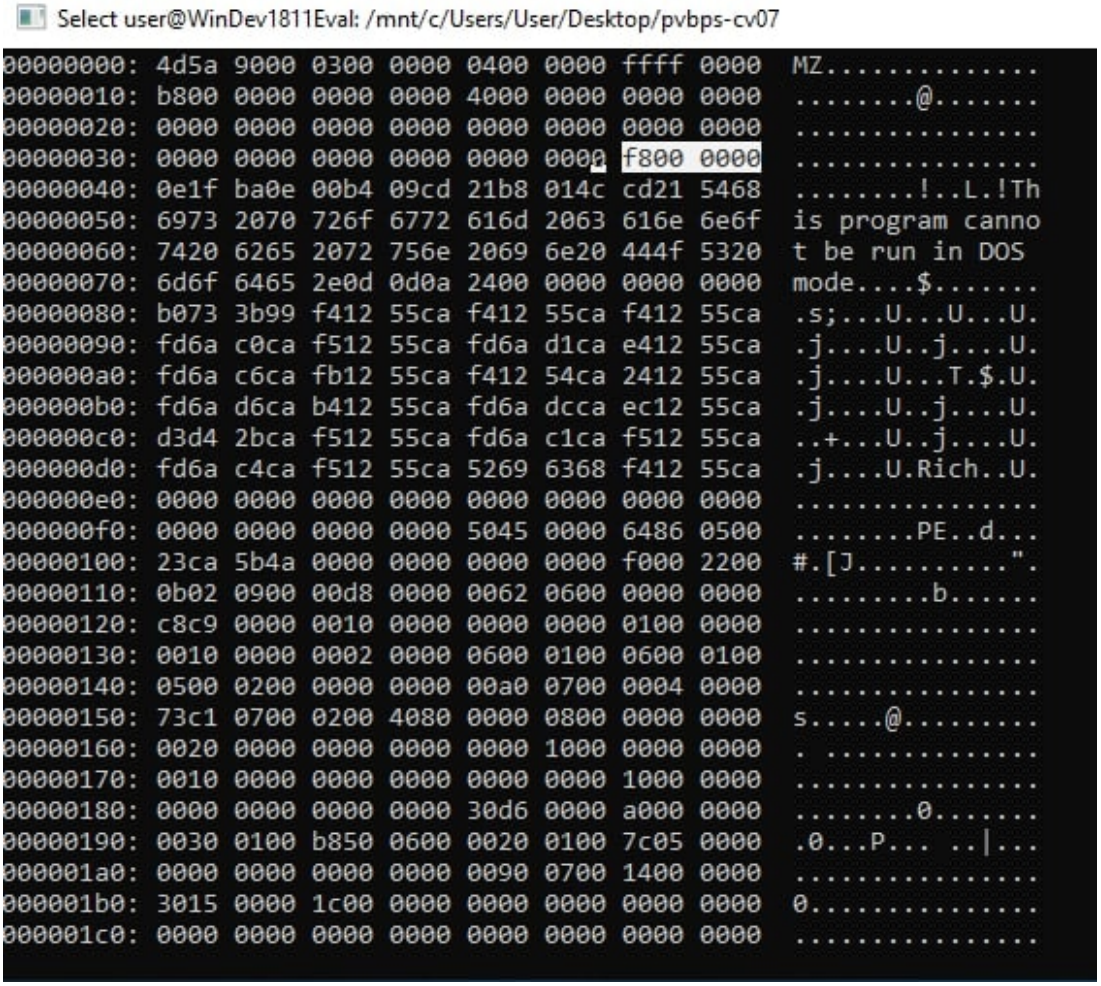
  Nějací zástupci:
  rar (samorozbalovací archiv WinRar sfx)
  Obsidium (spíš zaměřen na legitimní software)
  .netshrink pro .net aplikace

  pro javascript se spíše využívá pojem obfuskace/(minifikace), kdy se také takhle skrývá případný malware

2) Detailně se seznamte s hlavičkami užívanými u spustitelných souborů – především pak PE(COFF) a DOS hlavičkou. Stáhněte libovolný Hex editor.

- **Zjistěte, co znamená „e_lfanew“? Jaký má offset?**
  Spustitelné soubory ve Windows jsou složený z MS-DOS hlavičky a MS-DOS kódu, který je zde pouze pro zpětnou kompatibilitu a většinou vypíše hlášku, že tento program nelze spustit v DOS. Poté následuje PE hlavička obsahující informace o našem programu spolu s obsahem všech sekcí.
  Atribut **e_lfanew** je tedy adresa, kde začíná PE hlavička, protože MS-DOS kód má proměnlivou velikost. Tato položka je definována ve struktuře IMAGE_DOS_HEADER jako poslední atribut s offsetem 30*sizeof(WORD) = 0x3c.

- Pomocí Hex editoru otevřete malware, který Vám byl přidělen. Najděte/získejte hodnotu, která se nachází na poli „e_lfanew“. **PŘILOŽTE SCREENSHOT z Hex editoru, kde bude vidět nalezená hodnota.**
  V 32bitovém poli „e_lfanew“ na offsetu 0x3c je uložená hodnota 0xF8, jelikož jsou zde data uloženy ve formátu little-endian. Hodnota určuje, že PE hlavička začíná v souboru na offsetu 0xF8.



- V Hex editoru najděte místo, na které odkazuje hodnota z pole „e_lfanew“ (tedy nalezená hodnota z předchozího odstavce). **PŘILOŽTE SCREENSHOT z Hex editoru, kde bude vidět nalezená hodnota.**
  Na pozici 0xF8 v souboru začíná PE hlavička 16bitovým polem e_magic, které obsahuje vždy dva bajty 'PE' (0x50 0x45) za sebou v paměti.

Select user@WinDev1811Eval: /mnt/c/Users/User/Desktop/pvbps-cv07

```
00000000: 4d5a 9000 0300 0000 0400 0000 ffff 0000  MZ..............
00000010: b800 0000 0000 0000 4000 0000 0000 0000  ........@.......
00000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000030: 0000 0000 0000 0000 0000 0000 f800 0000  ................
00000040: 0e1f ba0e 00b4 09cd 21b8 014c cd21 5468  ........!..L.!Th
00000050: 6973 2070 726f 6772 616d 2063 616e 6e6f  is program canno
00000060: 7420 6265 2072 756e 2069 6e20 444f 5320  t be run in DOS
00000070: 6d6f 6465 2e0d 0d0a 2400 0000 0000 0000  mode....$.......
00000080: b073 3b99 f412 55ca f412 55ca f412 55ca  .s;...U...U...U.
00000090: fd6a c0ca f512 55ca fd6a d1ca e412 55ca  .j....U..j....U.
000000a0: fd6a c6ca fb12 55ca f412 54ca 2412 55ca  .j....U...T.$.U.
000000b0: fd6a d6ca b412 55ca fd6a dcca ec12 55ca  .j....U..j....U.
000000c0: d3d4 2bca f512 55ca fd6a c1ca f512 55ca  ..+...U..j....U.
000000d0: fd6a c4ca f512 55ca 5269 6368 f412 55ca  .j....U.Rich..U.
000000e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000f0: 0000 0000 0000 0000 5045 0000 6486 0500  ........PE..d...
00000100: 23ca 5b4a 0000 0000 0000 0000 f000 2200  #.[J..........".
00000110: 0b02 0900 00d8 0000 0062 0600 0000 0000  .........b......
00000120: c8c9 0000 0010 0000 0000 0000 0100 0000  ................
00000130: 0010 0000 0002 0000 0600 0100 0600 0100  ................
00000140: 0500 0200 0000 0000 00a0 0700 0004 0000  ................
00000150: 73c1 0700 0200 4080 0000 0800 0000 0000  s.....@.........
00000160: 0020 0000 0000 0000 0000 1000 0000 0000  . ..............
00000170: 0010 0000 0000 0000 0000 0000 1000 0000  ................
00000180: 0000 0000 0000 0000 30d6 0000 a000 0000  ........0.......
00000190: 0030 0100 b850 0600 0020 0100 7c05 0000  .0...P... ..|...
000001a0: 0000 0000 0000 0000 0090 0700 1400 0000  ................
000001b0: 3015 0000 1c00 0000 0000 0000 0000 0000  0...............
000001c0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

- Stáhněte libovolný program, který umí zobrazit hodnoty z PE hlaviček. **Vypište různé informace (dle Vašeho uvážení) z PE hlaviček o analyzovaném malware.**
  Pro analýzu byl využít program dumpbin /ALL Sample5 nebo objdump -x Sample5

- 100000000 image base (0000000100000000 to 0000000100079FFF)
  Sample5 je zkompilován s base adresou 0x1 0000 0000.
- C9C8 entry point (000000010000C9C8)
  první instrukce je na adrese 0x1 0000 C9C8
- 20B magic # (PE32+)
  jedná se o 64bitovou verzi aplikace
- 5 number of sections
  skládá se z 5 sekci:
  - 3000 .data - inicializované globální a statické proměnné
  - 1000 .pdat - metadata pro obsluhu vyjímek
  - 1000 .reloc - data pro relokaci programu na jiné místo ve virtuální paměti
  - 66000 .rsrc - ikonky, dialogy, menu, ...
  - E000 .text - instrukce programu
- 66200 size of initialized data (primárně v .rsrc sekci)
- 0 size of uninitialized data

program obsahuje 66200 bajtů inicializovaných dat/proměnných

- 1000 section alignment
  velikost sekce je vždy zaokrouhlena na horu na velikost stránky 4kB

3) Podívejte se blíže na tabulku importů, která bývá součástí spustitelných souborů.

- **K čemu slouží tato tabulka?**
  Adresy funkcí z DLL nejsou v době překladu známe, resp. mohou být odlišné pro různé verze DLL. Aby nemusel dynamický linker při startu programu opravovat všechny adresy programu, tak právě proto existuje tato tabulka. Tabulka pro každou použitou funkci z dll obsahuje instrukci skoku do této funkce. Dynamický linker tyto adresy při startu opraví. Aplikace následně místo volání funkce z knihovny volá funkci v tabulce, kde je skok na instrukci funkce z knihovny.

- Stáhněte program DependencyWalker (http://www.dependencywalker.com/) a zobrazte importované knihovny. **Vypište alespoň 3 knihovny, které jsou přímo importované analyzovaným malware:**
  KERNEL32.DLL
  USER32.DLL
  ADVAPI32.DLL
  GDI32.DLL

- **Z každé vybrané knihovny vypište alespoň jednu obsaženou metodu a dle dokumentace napište, co tato metoda dělá**:
  - **KERNEL32.DLL**
    - **GetTempFileNameA**
      Funkce vygeneruje cestu k dočasnému souboru, který je nutné následně vytvořit, malware si ze může ukládat např. stáhnutý EXE
      Soubor se nemaže, takže je nutné jej aplikačně smazat
    - **DeleteFileA**
      funkce smaže soubor. Malware může tuto funkci často používat pro zahlazení stop.
  - **USER32.DLL**
    - **ShowWindow**
      Funkce umožňuje například skrýt okno aplikace. Malware to může využít pro skryti sebe sama
  - **ADVAPI32.DLL**
    - **RegSetValueExA**
      Funkce nastavuje hodnotu v registrech. Malware si může například přidat AutoRun záznam, aby byl znovuspuštěn při startu. Případně si zde může odložit powershell skript zakódovaný v base64, což ale může být podezřelé už jen kvůli své velikosti.
  - **GDI32.DLL**
    - **GetDeviceCaps**
      - Funkce umožňuje získat informace o zařízení jako je například rozlišení displeje.