KATEDRA
INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

Cvičení 6 - Analýza malware

KMI/BEPS, KMI/BEZIT

2025/2026 - Letní semestr

Mgr. Jakub Juračka

Analýza malware

Analýza malware

- Malware většinou nemáme ve formě zdrojového kódu
  - $\Rightarrow$ pracujeme s binárními soubory
  - $\Rightarrow$ nevíme přesně, co program dělá
  - $\Rightarrow$ v případě ransomware, jedná se o "čestný" ransomware?
- Typické otázky:
  - Co malware dělá?
  - Jak se šíří?
  - Jak komunikuje?
  - Lze jeho chování zastavit / detekovat?

Analýza malware

Hlavní typy analýzy

- **Statická analýza**
  - bez spuštění programu
  - analýza binárky, stringů, struktur
  - rychlá a bezpečná

- **Dynamická analýza**
  - spuštění **v kontrolovaném prostředí**
  - sledujeme chování programu
  - realističtější, ale riziková

Analýza malware

Co hledáme ve statické analýze?

- **Instrukce (assembler)**
  - logika programu
- **Řetězce (strings)**
  - URL, IP adresy
  - názvy souborů
  - možné klíče
- **Struktura programu**
  - funkce, importy knihoven
  - odhad funkcionality

Analýza malware

Dynamická analýza

- Sledujeme běh programu:
  - přístup k souborům
  - změny v registru
  - vytváření procesů
- Síťová komunikace:
  - C&C servery
  - DNS, HTTP(S)
- Výhoda:
  - vidíme skutečné chování
- Nevýhoda:
  - malware může detekovat sandbox

Analýza malware

Ransomware – specifika

- Šifrování dat:
  - symetrické (rychlé)
  - asymetrické (bezpečné)
- Klíče:
  - lokálně
  - na serveru útočníka
- Realita:
  - zaplacení $\neq$ garance dešifrování

Analýza malware

Co s binárním souborem?

- Binární soubor lze otevřít i v textovém editoru
  - Obvykle jsou veškerá data nečitelná
- Lepší možností je hexaeditor
  - např.: Okteta, PSPad
  - `hexdump -C soubor`
- Soubory malware jsou obvykle distribuovány ve spustitelné formě → objektové soubory
  - Můžeme prozkoumat strukturu: `objdump -s soubor`
  - Přehledněji přes instrukce: `objdump -d -M intel soubor`
- Více informací v kurzu Operační systémy 1 od dr. Krajči:
  - `https://phoenix.inf.upol.cz/~krajcap/courses/2026LS/OS1/tutorial02.pdf`

Obrana malware

- Malware se aktivně brání analýze
  - odstranění debug symbolů
  - vložení „mrtvého“ kódu (neprováděné instrukce)
  - rozdělení důležitých dat (např. řetězců) na části + šifrování
- $\Rightarrow$ cílem je ztížit porozumění programu analytikem

Modifikace kódu

- Malware často mění svůj vlastní kód
  - **Polymorfismus**
    - mění se reprezentace kódu
    - funkcionalita zůstává stejná
  - **Metamorfismus**
    - mění se i struktura programu
- $\Rightarrow$ znemožnění detekce pomocí signatur (antiviry)
- Signatury jsou nejčastěji byte-sekvence, které jsou pro daný malware typické

Obfuskace

- Snaha o znepřehlednění kódu
  - zdrojového (typicky využíváno pro JavaScript)
  - binárního (assembler)
- Formy:
  - změna názvů proměnných a funkcí
  - vkládání zbytečných instrukcí
  - složité řízení toku programu
- $\Rightarrow$ kód je funkční, ale obtížně čitelný

Ukázka

Úkol

- Co dělá následující funkce v JavaScriptu? (vlastními silami) Popište váš postup, jak jste k řešení došli.

```
function MysteriousFunction(_0x32ad5b,_0x58440c,_0x439511){const
    _0x2c02fc=_0x58440c*_0x58440c-0x4*_0x32ad5b*_0x439511;if(
    _0x2c02fc>0x0){const _0x48d57c=(-_0x58440c+Math['sqrt'](
    _0x2c02fc))/(0x2*_0x32ad5b);const _0x4361ed=(-_0x58440c-Math['
    sqrt'](_0x2c02fc))/(0x2*_0x32ad5b);return[_0x48d57c,_0x4361ed];}
    else if(_0x2c02fc===0x0){const _0x487850=-_0x58440c/(0x2*
    _0x32ad5b);return[_0x487850];}}
```

- Stáhněte si binární soubor z adresy `https://apollo.inf.upol.cz/~janostik/slides/bezit/ransom`
  - Zkuste přijít na to, co program dělá.
  - **Pozor! Jedná se o ransomware, spusťe pouze na vlastní nebezpečí, můžete přijít o data!** (použít virtuální stroj?)
  - Jaké je heslo pro šifrování?
  - Je tento ransomware "čestný" a umožní i dešifrovat?