Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Využití nástrojů Windows Sysinternals pro efektivní analýzu operačního systému

Bakalářská práce

Autor: Martin Vahala

Studijní obor: Aplikovaná informatika

Vedoucí práce: Mgr. Josef Horálek Ph.D.

Hradec Králové 8/2022

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

……..

V Hradci Králové dne 14.8.2022 Martin Vahala

Poděkování:
Děkuji vedoucímu bakalářské/diplomové práce Mgr. Josefu Horálkovi Ph.D. za trpělivost při vedení práce a své rodině za dlouhodobou podporu během studií.

Anotace

Tato bakalářská práce se zabývá vybranými nástroji z procesní, bezpečnostní a informační skupiny balíku Sysinternals. Podrobně popisuje zobrazované informace, možnosti, nastavení, ovládání a kde je to možné, také grafické rozhraní jednotlivých nástrojů. V navazující praktické části lze nalézt nejprve návod k vytvoření virtuálního počítače a následně několik úloh, které dávají představu o možném využití vybraných nástrojů v praxi, a to nejen na místním, ale také vzdáleném počítači.

Klíčová slova: Sysinternals, operační systém Windows, analýza procesů, Process Explorer, Process Monitor, System Monitor, Autoruns

Annotation

Title: Using Windows Sysinternals for efficient analysis of the operating system

This Bachelor Thesis deals with selected tools from the process, security, and information group of the Sysinternals package. It describes in detail the information displayed, options, settings, controls, and the graphical interface of each tool where available. In the subsequent practical section, first instructions for creating a virtual machine can be found, followed by several tasks that give an idea of the possible use of the selected tools in practice, not only on a local but also on a remote computer.

Keywords: Sysinternals, the Windows operating system, process analysis, Process Explorer, Process Monitor, System Monitor, Autoruns

Obsah

Seznam obrázků

Všechny obrázky v této práci jsou původním dílem autora této práce.

1 Úvod

Nástroje Sysinternals jsou sadou více než 70 různých nástrojů pro analýzu operačního systému z rodiny Windows, které jsou používány profesionály z různých odvětví IT. Využití najdou nejen v případě nastalých problémů, ale i při běžné správě počítačových systémů.

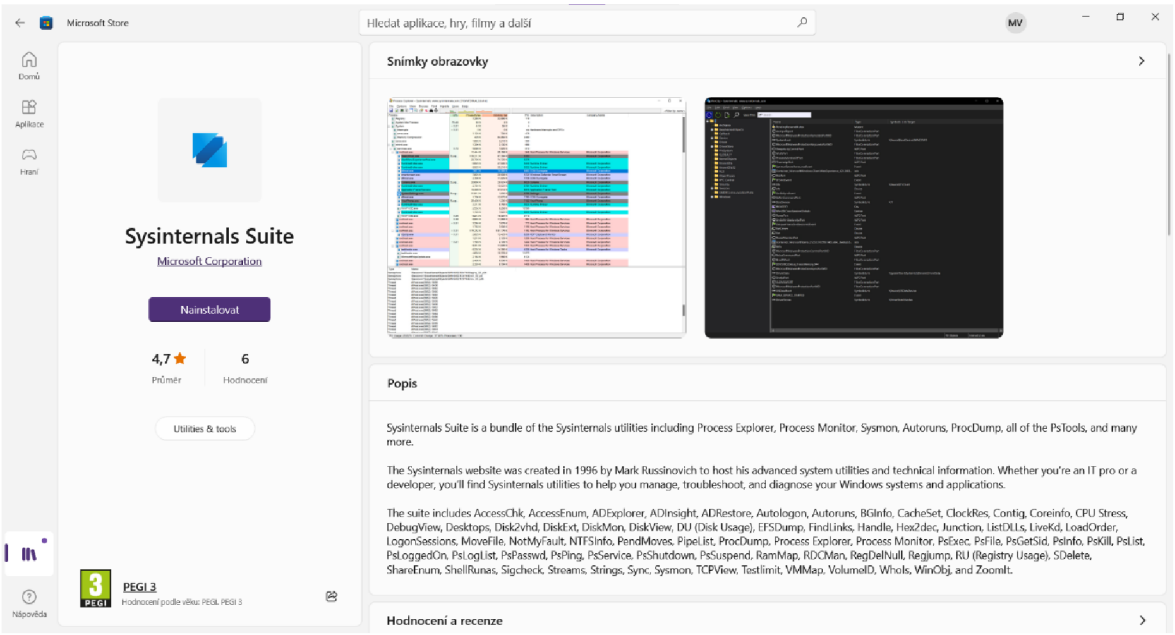
Autory prvních programů v tomto balíku byli Mark Russinovich a Bryan Cogswell, které časem doplnili další spoluautoři. Jak se můžeme dočíst v oficiální příručce pro tyto nástroje, Troubleshooting with the Windows Sysinternals Tools 2nd Edition [1-xxii] (pokračování knihy Windows Sysinternals Administrator‘s Reference), vůbec prvním nástrojem byl roku 1996 jednoduchý prográmek Ctrl2cap (mimochodem dodnes součást balíku Sysinternals Suite), který zajišťoval konverzi klávesy Capslock na levý control (Ctrl), protože právě na toto rozvržení klávesnice byl Mark Russinovich zvyklý z práce na UNIXových systémech. Vývoj v následujících letech zastřešovala firma Winternals Software, založená výše zmíněnými autory.

Firma Winternals Software (a s ní i práva na jejich produkty) byla následně v roce 2006 koupena Microsoftem, jehož zaměstnanci od té doby spolupracují na pokračujícím vývoji jednotlivých nástrojů a jejich přizpůsobování neustále se vyvíjejícím operačním systémům rodiny Windows (aktuálně Windows 11). Důsledkem této koupě bylo znepřístupnění zdrojového kódu nástrojů široké komunitě, nicméně programy samotné jsou neustále k dispozici zdarma.

Před popisem možností jednotlivých programů je nejprve třeba získat nástroje samotné. Zde máme několik možností:

- Všechny nástroje najednou si můžeme stáhnout v balíčku Sysinternals Suite, který si můžete stáhnout z části Downloads na stránkách Microsoft Docs [5], kdy celý balík má přibližně 44 MB.
- Pokud chceme využít jen konkrétní nástroj, můžeme si ho stáhnout z příslušné stránky na webu Microsoft Docs

- Od podzimu roku 2021 je rovněž možné stáhnout si celý balíček najednou z obchodu Microsoft Store (zde je ale nutné mít aktivní účet u společnosti Microsoft, založený např. při prvním spuštění nového PC). Součástí stránky Sysinternals na Microsoft Store je také záložka, kde si můžeme kontrolovat aktuálnost takto získaných nástrojů. Tímto způsobem získané nástroje jsou na PC nainstalovány (jako balíček, nelze si vybírat) a pro jejich odstranění je třeba je odinstalovat z Nastavení Windows – okno Aplikace a Funkce (navíc odinstalace je možná jen jako celek, aplikace Sysinternals Suite). Rovněž je třeba napsat, že každý z více než 70 nástrojů bude mít vlastní položku v menu Start a tím ho podstatně znepřehlední.



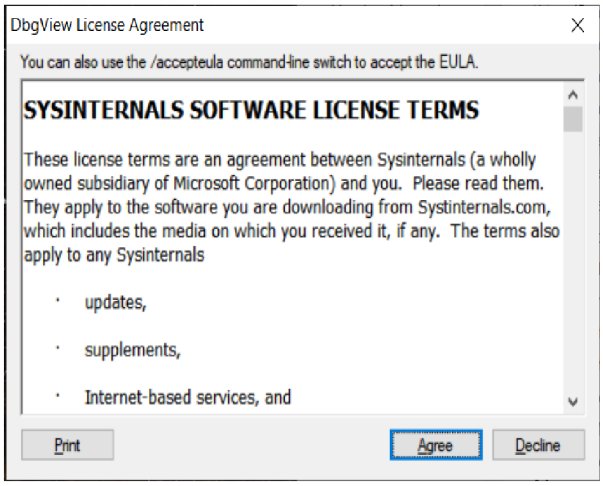
Obrázek 1 – Sysinternals Suite v Microsoft Store

- Nakonec je také možné libovolný nástroj rovnou spustit/kopírovat přímo z veřejně přístupného úložiště *Sysinternals Live* pomocí adresy \\live.sysinternals.com zadané do adresní řádky přímo v Průzkumníku souborů (Exploreru). Otevře se veřejně sdílená síťová složka, kde jsou všechny nástroje pravidelně aktualizovány.

Takto stažené programy není třeba nijak instalovat (s výjimkou monitorovacího nástroje SysMon a specifické funkce nástroje ProcDump), takže můžeme vybranou aplikaci rovnou spustit (u některých aplikací také pozor na architekturu vašeho OS – 32/64 bitů).

Některé nástroje lze pro specifická využití spustit i bez administrátorských oprávnění, nicméně většina z nich vyžaduje spuštění s právy administrátora systému (bude tedy třeba potvrdit UAC výzvu).

Při prvním spuštění jakéhokoli nástroje Sysinternals Suite je třeba nejprve odsouhlasit licenční podmínky (jejich plné znění najdete v příloze), ať už kliknutím na tlačítko Agree nebo zadáním argumentu -accepteula v případě, že program spouštíme z příkazové řádky (a nechceme, aby nám okno na obr. 2 vyskočilo).



Obrázek 2 – Potvrzení licenčního ujednání nástrojů Sysinternals

Nástroje Sysinternals jsou použitelné jak pro osobní verze operačního systému Windows (př. Windows 10), tak pro verze serverové (př. Windows Server 19) a existuje také verze pro architekturu ARM64 a nejnověji jsou také vytvářeny verze programu pro operační systém Linux.

Velké množství nástrojů vzniklo v době operačních systémů Windows 95 nebo 98, kdy tyto systémy nabízely jen omezené možnosti správy a konfigurace. A i když doba pokročila a konfigurační a analytické nástroje dodávané s operačními systémy jsou stále pokročilejší (jako příklad lze uvést obyčejný Správce úloh), najdou programy z balíku Sysinternals stále své uplatnění.

2 Cíl práce

Cílem této bakalářské práce je představit čtenáři vybrané nástroje ze skupiny Sysinternals, které jsou určené pro analýzu operačních systémů z rodiny Windows NT (ať už osobních, nebo serverových verzí).

V první (teoretické) části bude podrobně představeno několik autorem vybraných nástrojů z balíčku SysInternals Suite, konkrétně z podskupin Process Utilities, Security Utilities a System Information. Bude popsán jejich účel, detailní ovládání a možnosti, které jednotlivé nástroje nabízí (zpravidla půjde o přizpůsobení získávaných informací).

Ve druhé (praktické) části této práce najdete jednoduché praktické ukázky pro vyzkoušení možností těchto nástrojů. Tyto ukázky budou probíhat buď na fyzickém počítači nebo v prostředí virtuálního stroje, v obou případech vybavených systémem Windows 10 (součástí praktické části bude návod na vytvoření a nastavení daného VM). Budou připravené tak, aby šly kdykoli snadno reprodukovat – nebudou tedy vázané na konkrétní počítač a v něm instalované programy nebo přítomné soubory.

Takto získané zkušenosti pak může uživatel využít např. při práci systémového administrátora nebo testera/vývojáře software

3 Představení nástrojů Sysinternals

Nyní si popišme některé vybrané nástroje z balíku Sysinternals Suite. Nejprve si představíme procesní nástroje (Sysinternals Process Utilities), jako druhé bezpečnostní nástroje (Sysinternals Security Utilities) a nakonec nástroje pro získání informací o systému (System Information Utilities).

3.1 Sysinternals Process Utilities

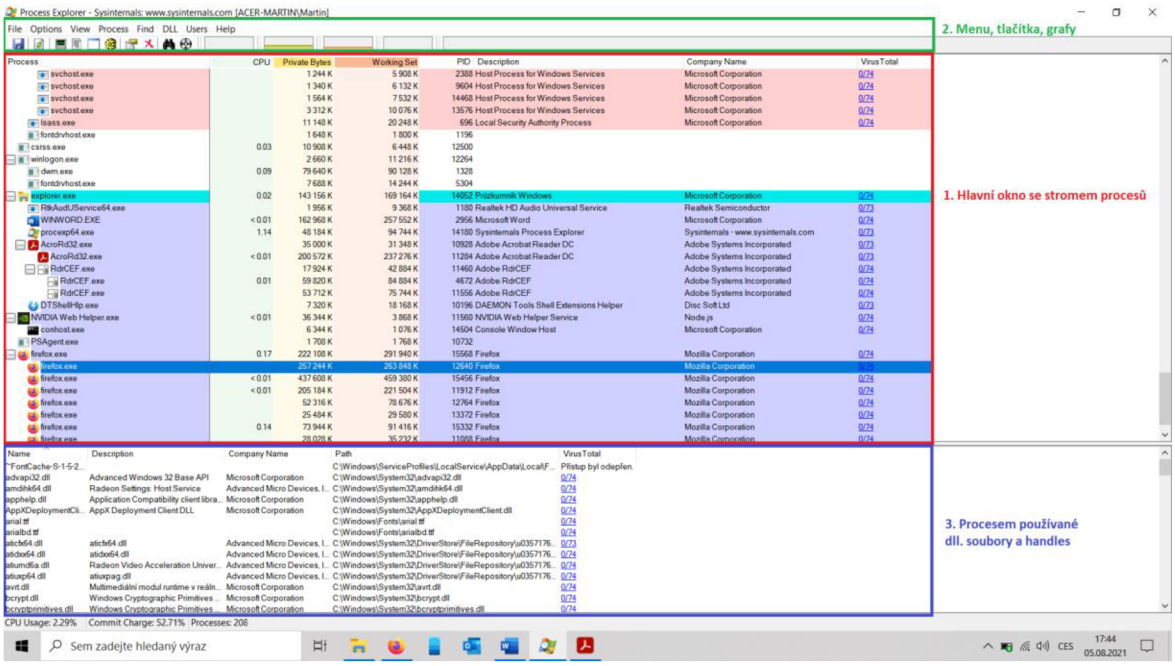
Do této skupiny nástrojů spadají některé nejkomplexnější a nejpoužívanější nástroje Sysinternals. Jde o níže představené **Process Explorer**, **Process Monitor** a **ProcDump**. Mimo ně bychom sem mohli zařadit CLI nástroje *Handle* a *ListDLL*, které stále nacházejí uplatnění při správě systému přes příkazový řádek nebo VMMap, sloužící pro analýzu paměti přidělené danému procesu.

3.1.1 Process Explorer

Nástroj Process Explorer, jedna ze součástí sady nástrojů Sysinternals Suite, je jednou z možných náhrad integrovaného Správce úloh v operačních systémech Microsoft Windows. Jeho první verze vznikla ve firmě Winternals Software LP Marka Russinoviche a Bryce Cogswella již v roce 2001 sloučením dvou v té době již existujících nástrojů (HandleEx a DLLView). Poskytne nám velmi podrobné informace o procesech běžících na spuštěné instanci operačního systému Windows.

Z námi získané sady Sysinternals Suite spustíme Process Explorer otevřením souboru procexp.exe pro 32bitovou verzi Windows, nebo procexp64.exe pro 64bitovou verzi operačního systému Windows (OS otevření nesprávné verze ani neumožní).

Po spuštění programu se otevře okno programu, rozdělené na několik částí (o každé se podrobněji dočtete dále):



Obrázek 3 – Hlavní okno s popisem částí

1. Na první pohled zaujme jeho hlavní část, kde můžeme vidět seznam spuštěných a barevně rozlišených procesů (včetně detailních informací o nich) ve formě přehledného, hierarchicky uspořádaného stromu.

2. V horní části vidíme standardní menu, několik tlačítek, fungující jako zkratky k některým funkcím programu a miniaturizaci grafu, který zobrazuje využití systémových prostředků v čase.

3. A nakonec okno ve spodní části, ve kterém se zobrazují informace o knihovnách (.dll souborech*) nebo o tzv. handle**, které proces, aktuálně vybraný v hlavním okně, využívá.

* *Dynamická knihovna (.dll – Dynamic Link Library) obsahuje kód, který může být použit vícero programy v jeden čas (v OS Windows např. knihovna comdlg32.dll zajišťuje dialogová okna a na ně navázané funkce, tato může být využívána* programy *pro Windows), což zajišťuje efektivnější práci se systémovými prostředky.* [2 – str.8]

** *Jako handle označujeme pomocný objekt bez známé vnitřní struktury, který je abstrakcí složitějšího objektu, spravovaného pomocí API (systém Windows 10 eviduje 53 různých typů objektů); do handle tabulky procesu je následně ukládána hodnota, která je pak předávána jako argument funkcím v API OS Windows, podle kterého dokáže systém vyhledat daný objekt, na který handle (jako index) odkazuje [3 – str. 143-145]*

1. Hlavní okno se stromem procesů a informacemi

V levé části okna vidíme strom rodičovských a synovských procesů, barevně odlišených a připravených k vybrání uživatelem. Vpravo od něj pak tabulku detailních informací o všech právě běžících procesech.

Poznámka: Kliknutím na libovolnou záložku nad tímto stromem lze měnit pořadí procesů v tabulce – místo implicitního stromu lze procesy seřadit např. podle abecedy, využití procesoru, id procesu a desítek dalších kritérií (o jejich nastavení čtěte dále). Přijdeme tím ovšem o přehledný strom, který je jednou z předností Process Exploreru

Implicitní nastavení po prvním spuštění programu nám bude zobrazovat následující informace:

1. Procentuální čas využití CPU daným procesem během posledního intervalu měření

2. Private Bytes ukazuje počet kilobajtů paměti přidělené procesu, která není sdílená s ostatními procesy, zahrnuje kombinaci paměti alokované na zásobníku i na haldě (stack/heap memory)

3. Working Set vypisuje množství fyzické paměti přidělené procesu Správcem paměti (Memory Managerem)

4. PID, tedy ID každého procesu

5. Description zobrazí základní informace o procesu, které Process Explorer dokázal získat (musí být spuštěn jako správce a mít validní cestu k umístění souboru)

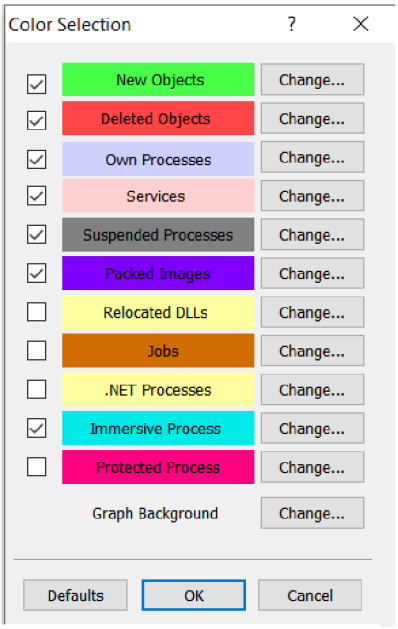
6. Company Name vypíše jméno společnosti stojící za vytvořením daného programu a jeho procesů (např. Microsoft Corporation apod.)

V nastavení programu je možné přidat desítky dalších sloupců s informacemi, o tomto v následující sekci popisující možnosti menu.

Jak již bylo řečeno, jednotlivé procesy ve stromě jsou barevně odlišeny. Tyto barvy nejsou vybrány nahodile a rozlišují nám jednotlivé typy procesů. Zde je jednoduchý přehled, co která barva znamená [1 – str.46]:

Opět vidíme, že ne všechno barvení je implicitně aktivní. Barvení aktivujete prostým zaškrtnutím v příslušném okně. Barvy přiřazené jednotlivým typům procesu lze samozřejmě měnit, prozatím si ale vystačíme s implicitním nastavením.

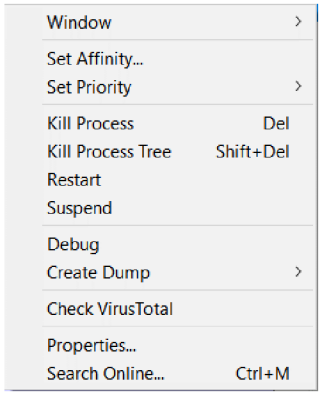
1. New Objects
   - Nově vytvořené procesy
2. Deleted Objects
   - Aktuálně ukončované procesy
3. Own Processes
   - Procesy spuštěné stejným uživatelem, jako Process Explorer
4. Services
   - Procesy obhospodařující Služby systému Windows (většinou půjde o hostovský proces svchost.exe)
5. Suspended Processes
   - Procesy, jejichž všechna vlákna byla pozastavena
6. Packed Images
   - Procesy, které by dle heuristické analýzy Process Exploreru mohly obsahovat zašifrovaný/komprimovaný spustitelný kód (je důrazně doporučeno tyto procesy zkontrolovat např. pomocí databáze VirusTotal – čtěte dále)



Obrázek 4 – Nastavení barev jednotlivým typům procesů

7. Immersive Process
    - Novinka od systému Windows 8 a dále, označuje procesy např. prostředí Metro ve Windows 8, widgety a dlaždicové aplikace

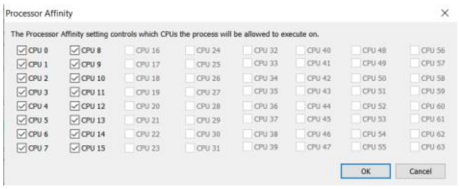
A nyní k akcím s jednotlivými procesy spojenými. Uživatel vybírá proces kliknutím levým tlačítkem myši, kamkoli v tabulce. Po vybrání procesu se v dolním okně zobrazí aktuálně využívané systémové knihovny nebo handles, dle nastavení programu. Kliknutím pravým tlačítkem myši na řádek procesu se otevře nabídka, které lze na daném procesu provést. Toto si opět ukažme na obrázku:



Obrázek 5 – Přehled akcí nad procesy

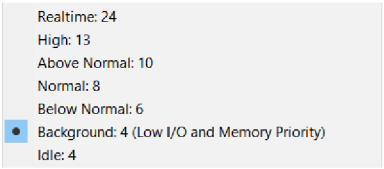
- **Window** – minimalizace/maximalizace okna aplikace, přesun okna do popředí apod (funkční jen pokud má proces okno k zobrazení)

- **Set Affinity** – zde lze nastavit, která z logických jader procesoru může daný proces využít



Obrázek 6 – Přiřazení jader procesoru procesu

- **Set Priority** – nastavení (a zobrazení) priority procesu, v 7 různých úrovních od 4 (nejnižší) po 24 (nejvyšší)



Obrázek 7 – Nastavení priority procesu

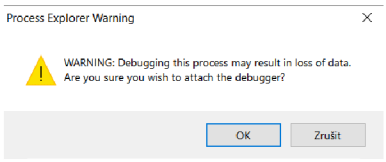
- **Kill Process / Kill Process Tree** – prvním příkazem můžete vynutit ukončení vybraného běžícího procesu, ve druhém případě dojde k ukončení celého stromu, který tento proces obsahuje (ukončen jak rodičovský, tak všechny synovské procesy); ukončení je implicitně nutno ještě potvrdit

    Pozn.: Příkaz Kill Process Tree je dostupný pouze tehdy, běží-li stromové zobrazení procesů. Nastavil-li si uživatel některou z možností řazení procesů, není tento příkaz aktivní.

- **Restart** – vynutí restartování procesu (opět nutné potvrdit)

- **Suspend** – pozastavení činnosti procesu (např. chceme-li dočasně uvolnit systémové zdroje)

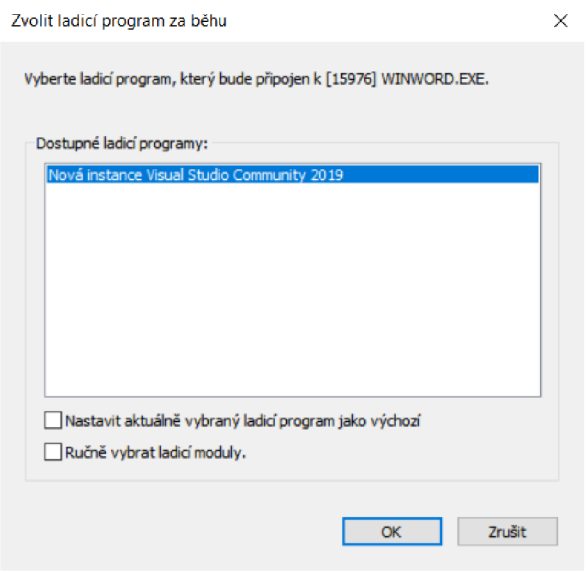
- **Debug** – možnost spustit a „nasadit“ na proces ladící program (též debugger). Po kliknutí nutno potvrdit varování.



Obrázek 8 – Debugger – varování

Jak uvádí Mark Russinovich [1 – str.55], tato položka nebude aktivní, není-li v registru systému Windows (adresa HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\AeDebug) zapsán některý z debuggerů.

Následně si uživatel zvolí některý z dostupných ladících programů a dále již pokračuje v jeho prostředí.



Obrázek 9 – Volba ladícího programu

- **Create Dump** – vytvoří .dmp soubor k pozdější analýze procesu např. přes nástroj WinDbg ze sady nástrojů Debugging Tools for Windows; uživatel má na výběr mezi vytvořením Minidump a Full Dump (kopie obsahu fyzické paměti vyhrazené procesu)

- **Check VirusTotal** – zkontroluje proces na webové stránce virustotal.com, která sdružuje desítky antivirových databází a vrátí (resp. obnoví) číslo říkající, v kolika těchto databázích je daný proces uveden jako škodlivý kód

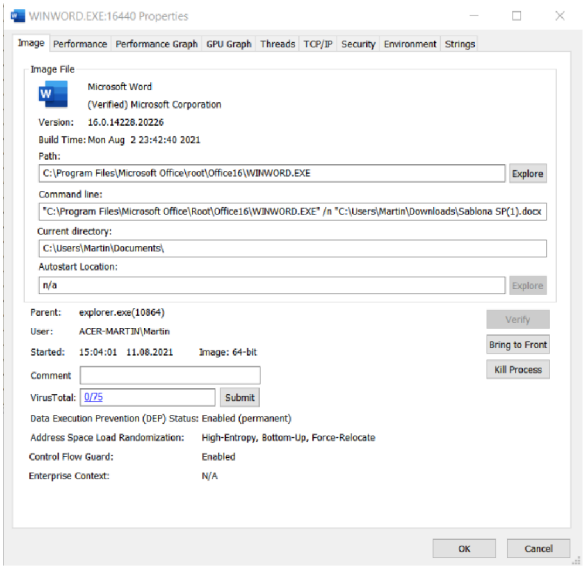
- **Search Online** – vyhledá název procesu uživatelovým webovým prohlížečem a vyhledávačem (př. Google)

- **Properties** – otevírá podokno s detailními informacemi o procesu (viz další stránky)

Okno Properties (Vlastnosti):

1. **Image**

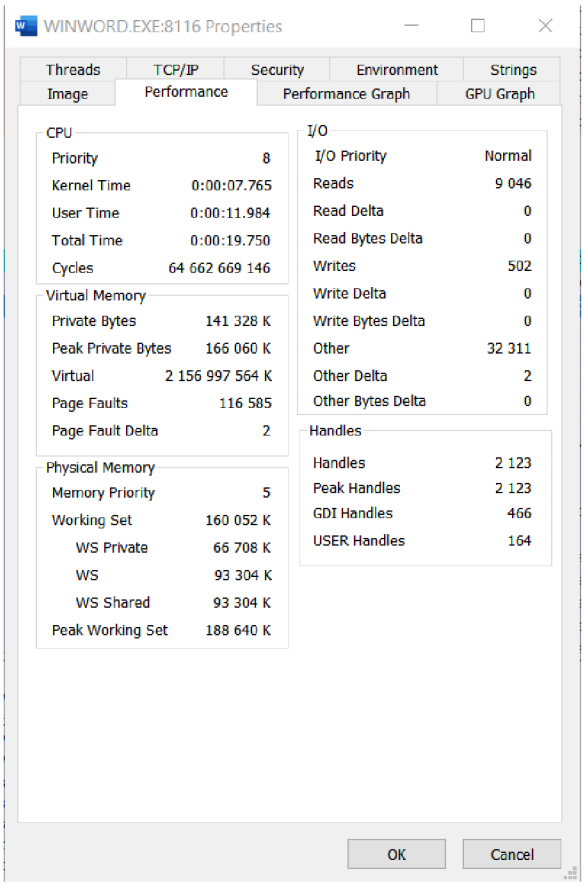
   Poskytuje základní získané informace o procesu – např. jméno, verzi programu, cestu k danému .exe souboru, rodičovský proces, jméno uživatele, který proces spustil, skóre v databázi VirusTotal.com, stav DEP (Data Execution Prevention) a další



Obrázek 10 – Properties – Záložka Image

2. **Performance**

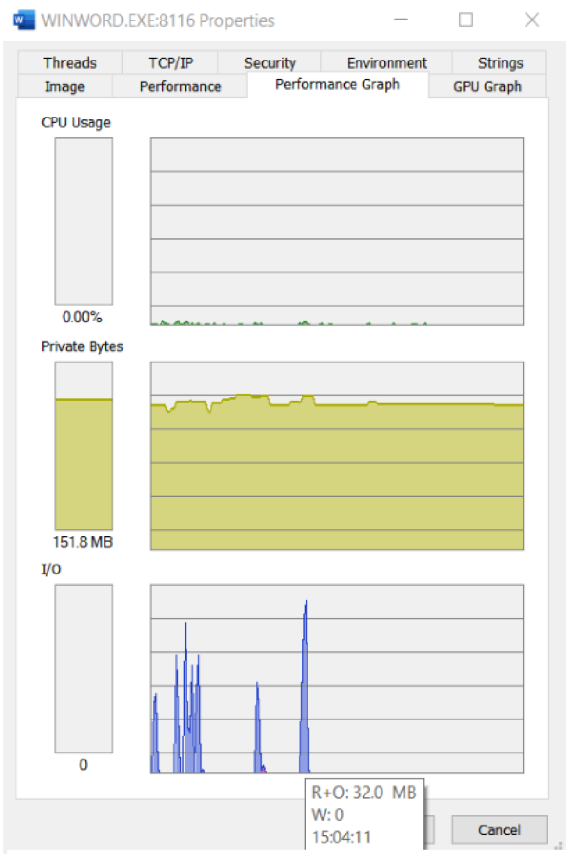
   Tato karta zobrazuje naměřené hodnoty systémových zdrojů (CPU, HDD, paměť – fyzická i virtuální), které daný proces využívá, kolik využil nejvíce, nastavení priorit pro daný proces a podobně.



Obrázek 11 – Properties – Záložka Performance

3. **Performance Graph**

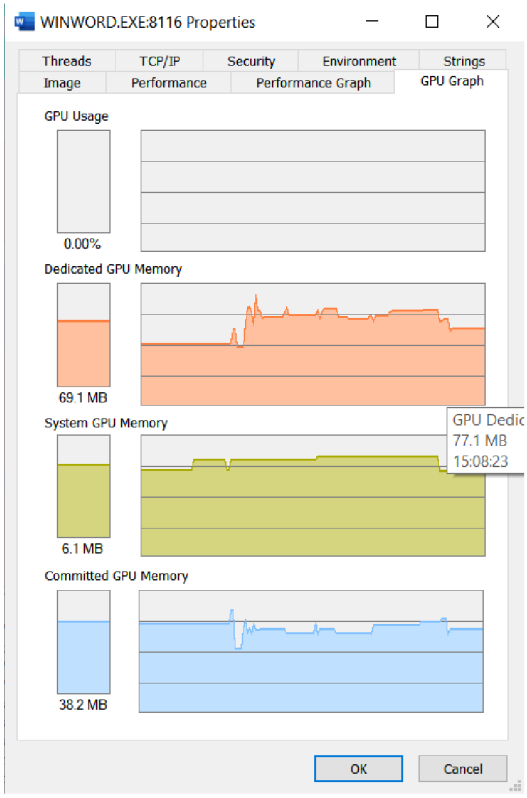
   Zobrazuje graf využití procesoru, paměti a disku daným procesem v čase (většinu těchto informací si lze také zobrazit v hlavním okně po adekvátním nastavení sloupců – viz dále).



Obrázek 12 – Properties – Záložka Performance Graph

4. **GPU Graph**

Grafické znázornění spotřebovávaných zdrojů grafické karty (grafických jader, dedikované paměti grafické karty a sdílené virtuální paměti).

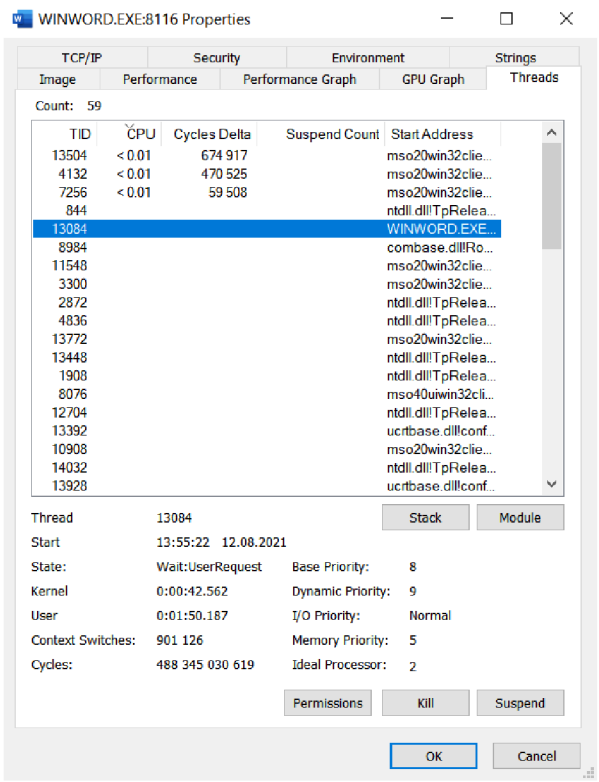


Obrázek 13 – Properties – Záložka s GPU grafy

5. **Threads**

Na této kartě můžeme vidět jednu z mnoha dalších předností nástroje Process Explorer, a to zobrazení detailních informací o jednotlivých vláknech daného procesu (který je považován za jakýsi kontejner sdružujícím přístup k systémovým zdrojům pro jednotlivá vlákna [2 – str.8]).

Údaje o vláknech v jednotlivých sloupcích zahrnují *TID* (thread ID – unikátní číselný identifikátor vlákna), *procentuální využití CPU*, *množství cyklů procesoru využitých procesem* mezi jednotlivými aktualizacemi informací (implicitně 1 s, nastavení viz dále), *služba systému Windows* využívaná procesem (na obrázku chybí) a *startovací adresou* vlákna*



Obrázek 14 – Properties – Záložka Threads

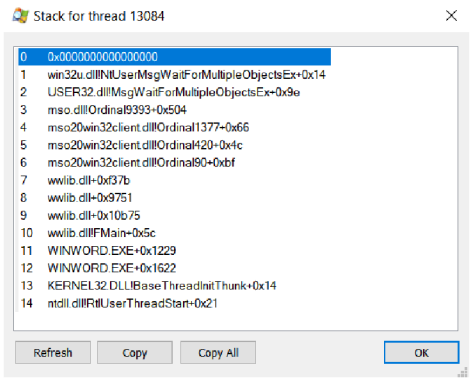
**Tuto adresu vysvětluje Mark Russinovich ve své knize* [1, str.97] přibližně takto:

Jde o symbolický název spojený s programem určeným místem ve virtuální paměti procesu, kde se vlákno začalo vykonávat. Název je uváděn ve formátu modul!funkce.

Pod tabulkou vidíme další doplňující informace, jako je přesný čas vytvoření vlákna, stav, ve kterém se vlákno právě nachází nebo priorita vlákna při přístupu k systémovým prostředkům.

Tlačítkem Stack otevřete nové podokno s výpisem ze zásobníku volání vlákna (Call Stack).

Tlačítko Module otevře podokno systému Windows – Vlastnosti, a to právě modulu, který dané vlákno využívá (většinou systémové .dll knihovny).

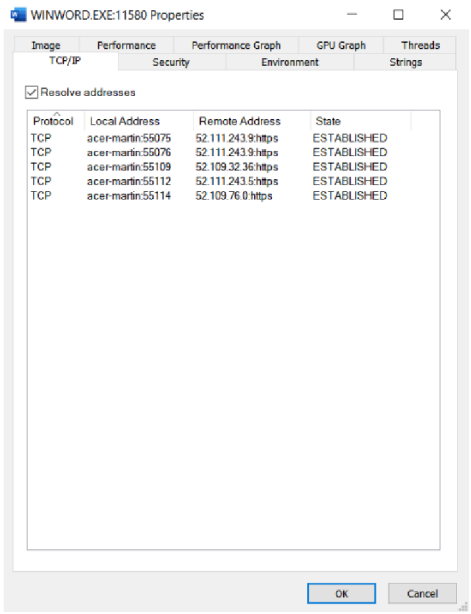


Obrázek 15 – Call Stack vlákna

6. **TCP/IP**

Jak je patrno již z názvu této záložky, najdeme zde seznam procesem vytvořených síťových spojení.

Ve sloupcích najdeme název použitého protokolu, místní IPv4 adresu (a použitý port), cílovou IPv4 adresu a stav spojení.

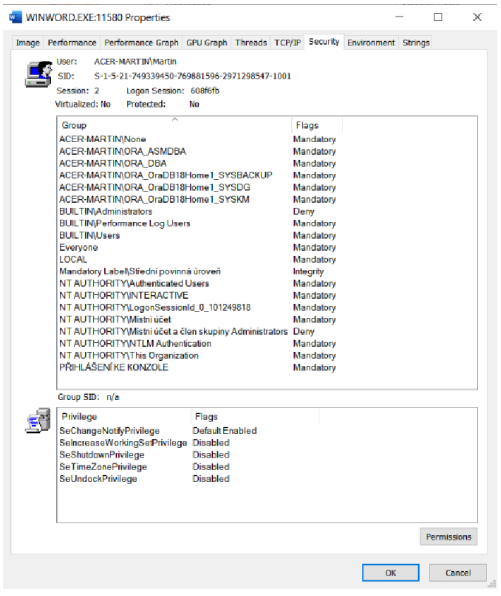


Obrázek 16 – Properties – Záložka TCP/IP

7. **Security**

Zde najdeme informace ohledně bezpečnosti provozu vybraného procesu. V první části vidíme uživatele, po kterým je proces spuštěn, unikátní identifikátor SID, stav virtualizace, UAC a další.

V tabulce ve druhé části podokna jsou vyjmenované skupiny uživatelů a jejich udělená oprávnění k danému procesu.



Obrázek 17 – Properties – Záložka Security

Kliknutí na tlačítko Permissions pak otevře standardní Windows podokno Zabezpečení, kde lze (s administrátorskými právy) měnit udělená oprávnění (Read, Write, Full Control apod.)

8. **Enviroment**

Na této záložce najdeme seznam „proměnných prostředí“ a jejich hodnoty.

Obrázek 18 – Properties – environment

9. **Strings**

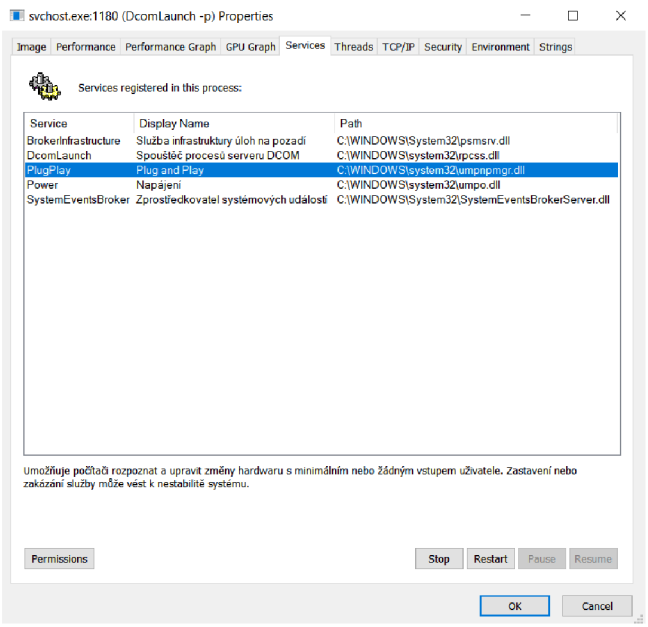
Zde jsou pouze vypsané textové řetězce (delší než 3 znaky), které Process Explorer dokázal najít.

10. **Services**

Tato záložka se zobrazuje pouze u procesů, které hostí některé ze Služeb systému Windows.

Jeden proces (svchost.exe) může obsahovat více než jednu službu.

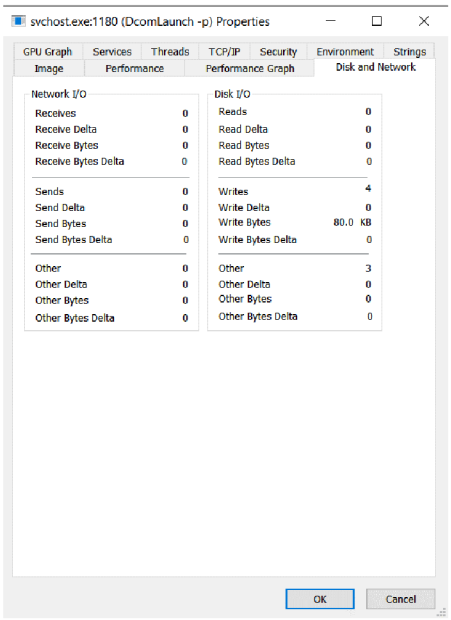
V tabulce pak lze nalézt jméno dané služby, cestu k souboru systémové knihovny a dole pak textový popis dané služby v implicitním jazyce systému Windows.



Obrázek 19 – Properties – services

11. **Disk and Network**

U některých procesů můžeme také najít záložku s detailními informacemi o aktivitách týkajících se zápisu/čtení z disku a síťového provozu



Obrázek 20 – Properties – Záložka disk & network

2. Menu, tlačítka

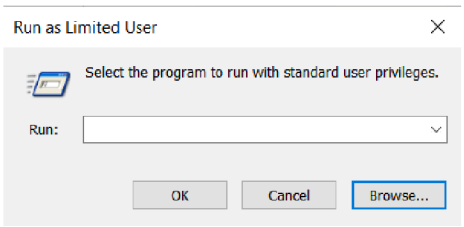
V této části si projdeme další možnosti nastavení programu Process Explorer v jeho menu a popíšeme rychlý přístup k některým funkcím přes dedikovaná tlačítka/ikony.

Menu

a. File

V první záložce menu je jen možnost otevřít standardní windowsovské okno Spustit, dále tu lze uložit výpis z programu (ve formátu .txt), lze odtud ovládat samotný počítač (odhlásit uživatele, uspat nebo i vypnout PC) a ukončit program Process Explorer.

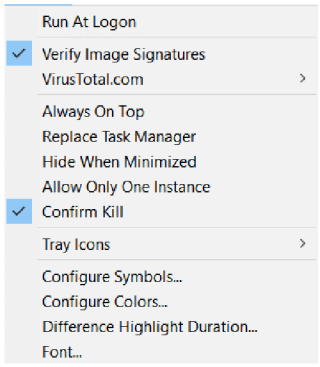
Za zajímavost lze pokládat možnost Run as Limited User, tímto příkazem má administrátor možnost spustit program s omezenými právy běžného uživatele [4, str.96].



Obrázek 21 – File – Run as Limited User

b. Options

Vedle možností jako je nastavení spouštění Process Exploreru hned po přihlášení uživatele k systému nebo aktualizace databáze VirusTotal.com nabízí tato záložka několik dalších možností._Tou nejzajímavější je možnost nastavit si Process Explorer jako trvalou náhradu za Task Manager.



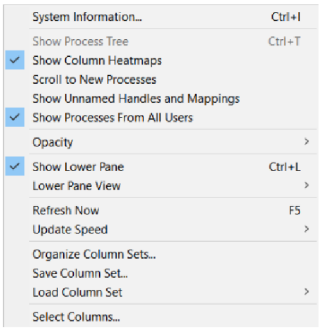
Obrázek 22 – Menu – options

Poznámka: Tato změna (odehrávající se v registrech systému Windows) je společná pro všechny uživatele daného počítače. Pokud však administrátor umístí soubor procexp.exe / procexp64.exe mimo přístup ostatních uživatelů PC (tedy jinam než do některé ze sdílených složek), nebudou tito mít možnost ho spustit a zároveň nepůjde spustit Task Manager (klávesovými zkratkami, z nabídky Start ani přes Win+R). Na toto je tedy nutno brát zřetel. [1, str.109]

Dále tu lze nastavit minigrafy (Tray Icons), které budou při běhu programu zobrazeny na Hlavním panelu Windows

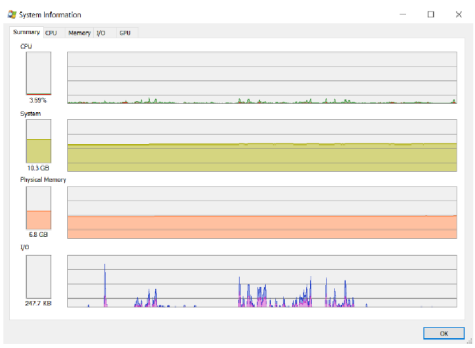
c. View

Záložka obsahující velice zajímavé informace o systému a možnosti nastavení Process Exploreru.



Obrázek 23 – Menu – záložka view

Hned první položka – System Information – po rozkliknutí zobrazí podokno s grafy zobrazujícími aktuální využití systémových prostředků.



Obrázek 24 – System Information – Summary

- Nejprve se zobrazí celkový přehled (**Summary**) o využití CPU, celkové paměti systému, fyzické RAM paměti a využití disku.

- Na druhé záložce (**CPU**) vidíme graf využití CPU (implicitně celkový, přepínačem ve spodní části lze zobrazit grafy ke každému logickému procesoru zvlášť) a některé informace o aktuálním využívání CPU

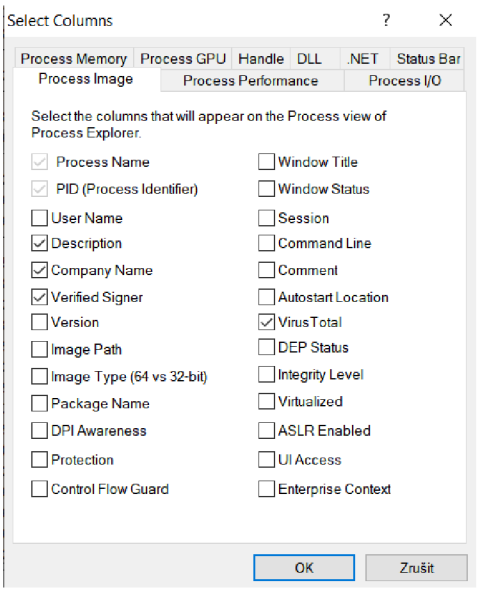
- Třetí záložka (**Memory**) zobrazuje (vedle grafů totožných s těmi v záložce Summary) detailní informace o využívání systémové a fyzické paměti.

- Čtvrtá záložka (**I/O**) poskytuje graf a některé informace o využití disku.

- A konečně pátá záložka (**GPU**) ukazuje historii využití grafické karty a její dedikované i sdílené paměti

Dále lze v této záložce nastavit frekvenci obnovování informací o procesech (položka **Update Speed**). Nejvyšší frekvence je obnovení každých 0.5 sekundy, nejnižší po 10 sekundách (samozřejmě je nutno počítat s tím, že čím vyšší frekvence, tím větší zatížení procesoru). Implicitně je nastaveno obnovování po 1 sekundě.

Zcela na konci této nabídky je pak klíčová položka **Select Columns**, která nám umožňuje zvolit si, které informace o procesech chceme zobrazit v hlavním okně programu. Lze tu zaškrtnout ke zobrazení informací přes 100 sloupců v několika různých kategoriích, zájemci si mohou podrobné informace přečíst v knize Troubleshooting with the Windows Sysinternals Tools od Marka Russinoviche [1, str. 55–68]



Obrázek 25 – Select Columns

d. Process

Možnosti záložky Process jsou totožné jako nabídka po kliknutí na proces v tabulce pravým tlačítkem myši, toto bylo detailně popsáno výše.

e. Find

V této záložce najdeme jedinou možnost, a to **Find Handle or DLL.** Po kliknutí se otevře okno, kam lze zadat část textového řetězce identifikujícího DLL nebo Handle.

Toto má praktické využití např. v dobře známé situaci, kdy Windows odmítá „bezpečně odpojit“ USB disk kvůli tomu, že je stále používán. Jako textový řetězec pak stačí zadat písmeno dané jednotky (př. D:\), potvrdit a program vypíše procesy, které používají .dll soubory nebo handles s daným popisem. Následně tyto stačí ukončit (Kill Process).

f. DLL

Když si ve spodním podokně vyberete systémovou knihovnu, zde k ní můžete získat detailní informace (Properties), vyhledat její název online nebo zkontrolovat v databázích VirusTotal.com (tedy totéž co získáte kliknutím na daný .dll soubor ve výše zmíněném podokně pravým tlačítkem myši)

g. Users

Tato záložka umožňuje základní ovládání spojené s k PC přihlášenými uživateli – lze je odhlásit, poslat zprávu jednotlivým uživatelům atd.

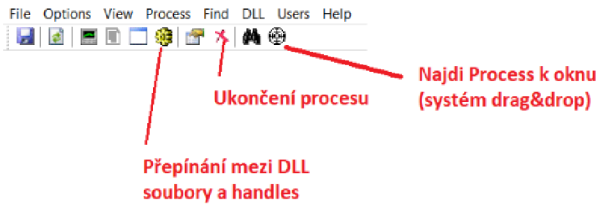
h. Help

Odtud lze otevřít základní nápovědu k programu.

Tlačítka

Pod lištou menu najdeme tlačítka k ovládání vybraných funkcí Process Exploreru. Většina již byla zmíněna (vyvolání Systém Information, Refresh, Kill Process).

Za zmínku stojí tlačítko k **přepínání zobrazení mezi .dll knihovnami a handle objekty** používanými označeným procesem nebo tlačítko k označení procesu, který používá označené okno*



Obrázek 26 – Tlačítka funkcí

Pozn. Funguje stylem drag&drop – uživatel „přetáhne“ myší z tlačítka do otevřeného okna, ke kterému chce najít proces.

3. DLLs a handles

Zde jen krátce k panelu, který lze najít ve spodní části hlavního okna programu (viz úvodní obrázek k Process Exploreru). Podle nastavení v menu programu se zde zobrazují buď systémové knihovny (.dll), které proces používá, jednotlivé handles (abstrakty souborů otevřených programem) nebo vlákna, která daný proces obsahuje. Pravé kliknutí na vybranou knihovnu/handle rozbalí stejnou nabídku, jako položka DLL v menu (popsána výše).

Za zajímavost lze považovat možnost prohlédnout si v tomto panelu všechny aktuálně načtené knihovny – stačí si nastavit zobrazování .dll souborů a následně ve stromě procesů najít rodičovský proces **System (PID 4)** a označit ho – do daného panelu pak dojde k vypsání nalezených knihoven a ovladačů načtených v paměti počítače [1-str.74].

Name	Description	Company Name	Path	Virus Total	Verified Signer
cambriat.ttf			C:\Windows\Fonts\cambriat.ttf	0/73	(Verified) Microsoft Windows
cfgmgr32.dll	Configuration Manager DLL	Microsoft Corporation	C:\Windows\System32\cfgmgr32.dll	0/70	(Verified) Microsoft Windows
clbcatq.dll	COM+ Configuration Catalog	Microsoft Corporation	C:\Windows\System32\clbcatq.dll	0/71	(Verified) Microsoft Windows
cldapi.dll	Cloud API user mode API	Microsoft Corporation	C:\Windows\System32\cldapi.dll	0/70	(Verified) Microsoft Windows
combase.dll	Microsoft COM for Windows	Microsoft Corporation	C:\Windows\System32\combase.dll	0/72	(Verified) Microsoft Windows
combase.dll.mui	Microsoft COM for Windows	Microsoft Corporation	C:\Program Files\WindowsApps\MicrosoftLanguageExperienceP...	Unknown	(Verified) Microsoft Corporation
comctl32.dll	User Experience Controls Library	Microsoft Corporation	C:\Windows\WinSxS\amd64_microsoft.windows.common-controls...	0/70	(Verified) Microsoft Windows
comdlg32.dll	Common Dialogs DLL	Microsoft Corporation	C:\Windows\System32\comdlg32.dll	0/73	(Verified) Microsoft Windows
comdlg32.dll.mui	Common Dialogs DLL	Microsoft Corporation	C:\Program Files\WindowsApps\MicrosoftLanguageExperienceP...	Unknown	(Verified) Microsoft Corporation
coml2.dll	Microsoft COM for Windows	Microsoft Corporation	C:\Windows\System32\coml2.dll	0/72	(Verified) Microsoft Windows
CompPkgSup.dll	Component Package Support DLL	Microsoft Corporation	C:\Windows\System32\CompPkgSup.dll	0/73	(Verified) Microsoft Windows
CONTAB32.DLL	Outlook Address Book Service	Microsoft Corporation	C:\Program Files\Microsoft Office\root\Office16\CONTAB32.DLL	0/73	(Verified) Microsoft Corporation
CoreMessaging.dll	Microsoft CoreMessaging Dll	Microsoft Corporation	C:\Windows\System32\CoreMessaging.dll	0/73	(Verified) Microsoft Windows
CoreUIComponents...	Microsoft Core UI Components Dll	Microsoft Corporation	C:\Windows\System32\CoreUIComponents.dll	0/71	(Verified) Microsoft Windows
cour.ttf			C:\Windows\Fonts\cour.ttf	0/73	(Verified) Microsoft Windows
crypt32.dll	Crypto API32	Microsoft Corporation	C:\Windows\System32\crypt32.dll	0/70	(Verified) Microsoft Windows
crypt32.dll.mui	Crypto API32	Microsoft Corporation	C:\Program Files\WindowsApps\MicrosoftLanguageExperienceP...	Unknown	(Verified) Microsoft Corporation
cryptbase.dll	Base cryptographic API DLL	Microsoft Corporation	C:\Windows\System32\cryptbase.dll	0/71	(Verified) Microsoft Windows

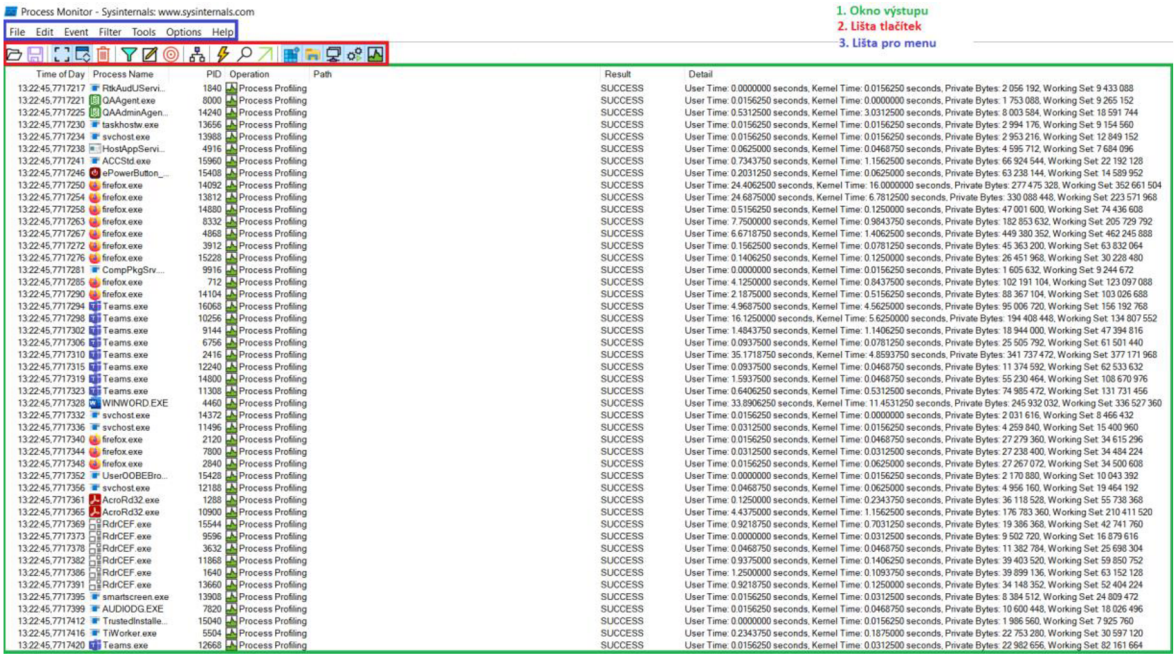
Obrázek 27 – Ukázka načtených .dll knihoven procesu WINWORD.exe

3.1.2 Process Monitor

Zatímco výše zmíněný Process Explorer ukazuje jakýsi statický obraz zobrazující stav operačního systému v dané chvíli, Process Monitor, vzniklý sloučením nástrojů FileMon a RegMon, nám umožňuje sledovat a zaznamenat chování uvnitř našeho OS během určitého časového úseku. Toto chování zahrnuje sledování interakcí mezi procesy (až na úroveň jednotlivých vláken) a systémovými prostředky, které nám náš OS nabízí. Tímto je myšlen souborový systém, síťová připojení, registry Windows a podobně. Také je třeba podotknout, že na rozdíl od Process Exploreru nelze pomocí Process Monitoru jednotlivé procesy ovlivňovat (ukončovat, pozastavovat apod.).
Tyto interakce probíhají velmi rychle a ve velkém množství, proto Process Monitor nabízí možnost poskytované informace efektivně filtrovat a zobrazit si pouze ta data, která jsou pro nás v danou chvíli podstatná (okno pro filtrování se zobrazí automaticky při startu programu nebo ho lze kdykoli manuálně vyvolat, viz dále).

Pozn.: Selecký [4] na str. 116 udává, že aplikace Process Monitor ke svému spuštění nevyžaduje administrátorská oprávnění. To je v rozporu s upozorněním Marka Russinoviche, autora nástroje a knihy Troubleshooting with the Windows Sysinternals Tools [1], který naopak upozorňuje, že aplikace k zachycení jednotlivých událostí využívá ovladač jádra (kernel driver) a proto ke svému chodu naopak administrátorská práva (tedy potvrzení Windows UAC) vyžaduje. Bez nich lze pomocí programu pouze zobrazit již dříve získaná, ve vhodném formátu uložená data (viz dále), a to pouze pomocí využití příkazové řádky k omezenému ovládání programu [1 – str.146]. Ke stejným závěrům došel i autor této práce.

Na obrázku 28 lze vidět příklad hlavního okna a jeho 3 části (barevně odlišeny):



Obrázek 28 – Přehled hlavního okna Process Monitoru

1. **Okno pro výstup**

Ihned po svém startu začne program zaznamenávat události, které v systému Windows nastaly, a vybrané údaje o nich zapisuje právě do tohoto okna.

Zaznamenané události můžeme rozdělit do 5 základních skupin (tříd):

I. Operace související s registry: tvorba klíčů, úprava a čtení jejich hodnot
II. Manipulace se souborovým systémem – vytváření souborů, jejich čtení, zápis a s tímto související řešení synchronizace – uzamčení a opětovné odemčení souborů
III. Síťové operace – záznamy o zprávách odeslaných a přijatých síťovými protokoly TCP a UDP – k zachycení využívá systémový nástroj ETW (Event Tracing for Windows)
IV. Aktivity na úrovni procesů a vláken – tvorba procesů, vláken a jejich ukončování, operace s .dll knihovnami nebo ovladači zařízení
V. Profilování procesů – události zachycující informace o procesech (využití paměti, čas v uživatelském/kernel módu, přepínání kontextu apod.) a také záznamy událostí z nástroje ProcDump (více v kapitole 3.1.3)

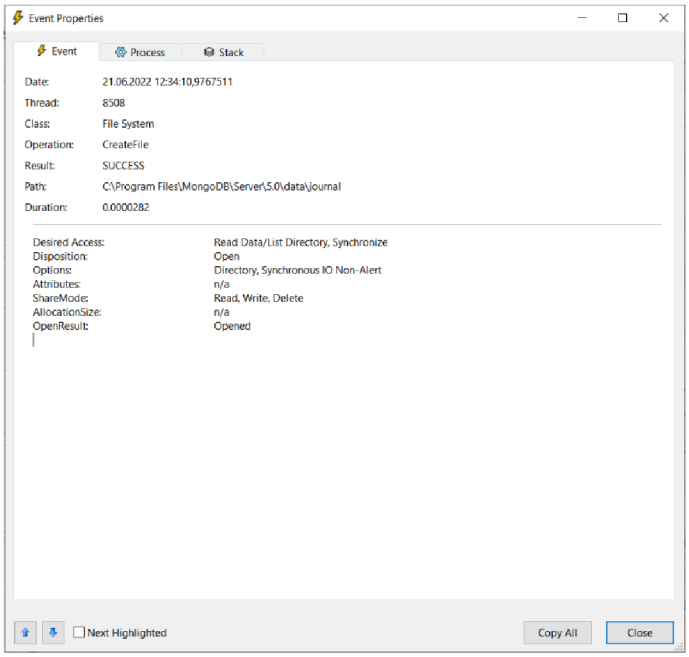
Každý řádek výstupu představuje zaznamenanou událost, ke které jsou v příslušných sloupcích zobrazena příslušná data. Implicitně se zobrazuje toto:

a) Time of Day – čas zachycení události s přesností na sedm desetinných míst (hh:mm:ss.xxxxxxx)
b) Process Name – jméno procesu
c) PID – identifikátor procesu
d) Operation – některý z výše zmíněných typů provedené operace
e) Path – typicky cesta do registru nebo k souboru
f) Result – výsledek dané operace, např.:
    - SUCCESS
    - ACCESS DENIED
    - PATH NOT FOUND
    - BUFFER OVERFLOW
    - … a další

g) Detail – další detaily k danému záznamu

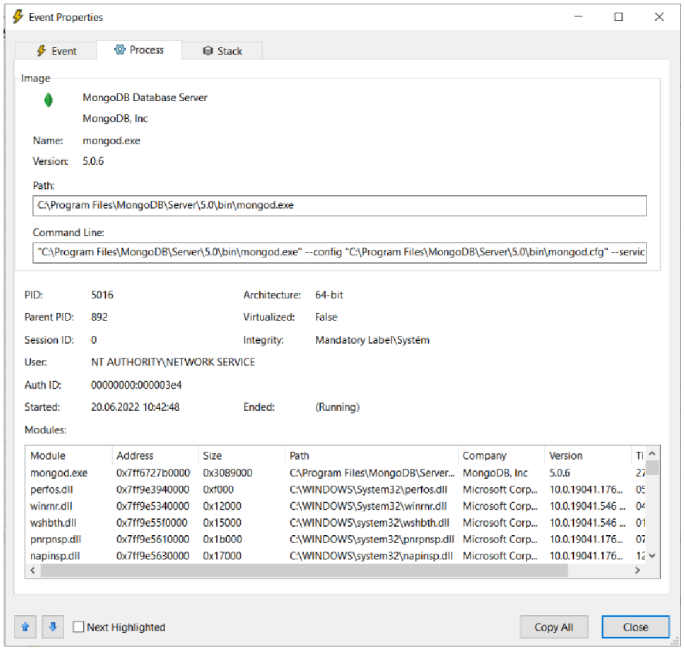
Dvojklikem na řádek (představující zachycenou událost) se otevře podokno s detaily o dané události a třemi záložkami – Event, Process a Stack.

V záložce **Event** vidíme informace týkající se okolností záchytu dané operace – kdy k ní došlo, TID, zařazení operace mezi pětici výše zmíněných tříd (operace se souborovým systémem, s registry, sítí apod.), typ provedené operace (vytvoření souboru, jeho uzamčení, tvorba vlákna, tvorba klíče v registrech, …), její výsledek, cestu k danému adresáři/záznamu v Editoru Registru a čas, který byl třeba k provedení operace v dané události.



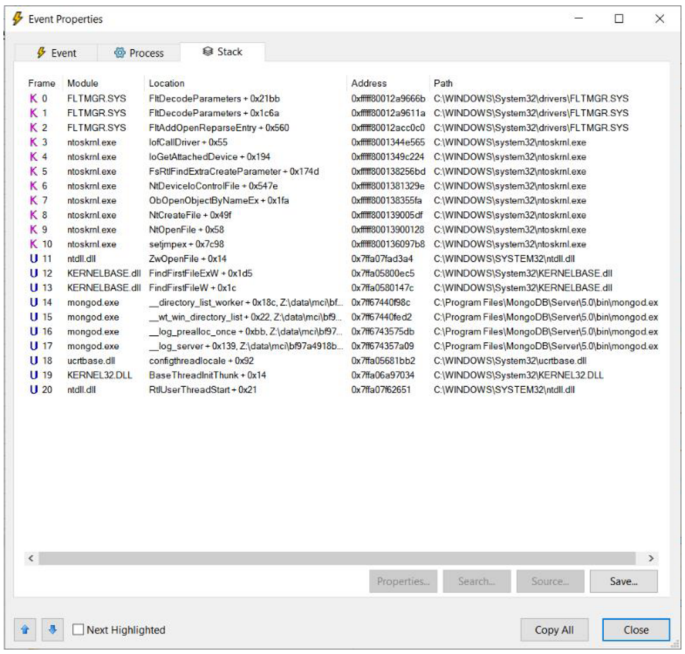
Obrázek 29 – Detail události – záložka Event

Druhou záložkou je **Process**, kde můžeme najít detaily k procesu, který danou událost vyvolal – tedy k procesu, který provedl danou operaci. Půjde např. o název a verzi aplikace, jejího výrobce, ID, datum a čas vytvoření daného procesu a seznam načtených modulů, které daný proces používá (totožný výpis jako nám zobrazí tlačítko Module v záložce Threads u detailu procesu v Process Exploreru – více na straně 13).



Obrázek 30 – Detail události – záložka Process

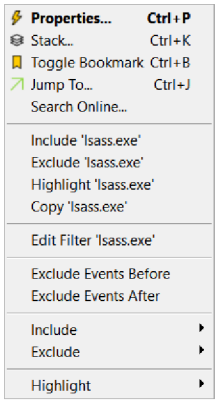
Poslední záložkou je **Stack**, kde můžeme vidět výpis ze zásobníku volání daného procesu. Jednotlivé záznamy rozlišují, zda šlo o volání funkce v kernel nebo uživatelském módu, který modul provedl dané volání, místo v daném modulu, které volalo funkci (ve formátu %název funkce + offset uvnitř kódu%), odkaz do virtuálního adresního prostoru a cestu k umístění daného modulu na disku.



Obrázek 31 – Detail události – záložka Stack

Kombinace těchto informací umožňuje vývojářům softwaru zjistit, kde při běhu programu nastává problém a ten následně odstranit.

Kliknutím pravým tlačítkem myši na řádek otevřete nabídku s vybranými funkcemi (totožnou jako nabídka Event v menu programu). Položky Properties a Stack otevřou detail události (viz předchozí stránka), lze zde vytvořit záložku ve výpisu, otevřít adresář odpovídající záznamu ve sloupci Path u dané události a vyhledat název daného procesu pomocí vyhledávače nastaveného jako primární ve vašem internetovém prohlížeči (typicky půjde o Google vyhledávač).



Obrázek 32 – Rychlá nabídka

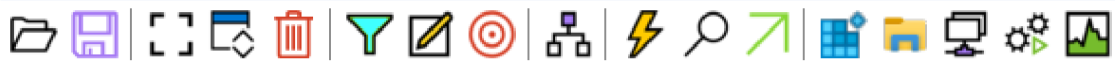
Dále je zde možnost rychle vytvořit filtr odpovídající dané události. Je třeba podotknout, že tato nabídka bere v potaz, do kterého sloupce dané události bylo pravým tlačítkem kliknuto. Místo názvu procesu (na obrázku 32 „lsass.exe“) zde tedy může být cokoli co bylo nastaveno k zobrazení k dané události (třeba ID procesu, typ operace, uživatel a podobně).

Položka Edit Filter otevře podokno ke správě aktivních filtrů výpisu, o tom dále v kapitole o menu programu. Položkami Exclude Events Before a After lze provést rychlý ořez výpisu událostí.

Pozn.: V době psaní této bakalářské práce položky include, exclude ani highlight nebyly funkční, jejich funkce pravděpodobně převzaly jiné položky v menu, hlavně v záložce Filter a jsou pravděpodobně určeny k odstranění.

2. Lišta tlačítek

Ikony na liště tlačítek slouží k rychlému použití vybraných funkcí Process Monitoru. Tyto si podrobně představíme v další části pojednávající o funkcích menu.



Obrázek 33 – Lišta tlačítek Process Monitoru

Postupně zleva jde o:

- Open – možnost otevření dříve vytvořeného záznamu událostí (pouze soubory s příponou .pml)
- Save – uložení aktuálního výpisu událostí (nachází se v okně pro výstup) do souboru pro pozdější analýzu (ve formátu .pml, .csv nebo .xml)

- Capture – pozastavuje a spouští záchyt událostí
- Autoscroll – posun okna s každou další zachycenou událostí
- Clear – vymaže doposud zachycené události v hlavním okně

- Filter – otevření nabídky pro nastavení filtrování událostí (podrobnosti viz dále)
- Highlight – barevné podbarvení záznamů odpovídajících nastavenému filtru
- Include process from window – podobně jako u Process Exploreru, můžeme přetáhnutím (drag and drop) do otevřeného okna vyfiltrovat záznamy týkající se daného procesu

- Process Tree – otevírá podokno stromu procesů

- Event Properties – podobně jako dvojklikem do výpisu událostí, toto tlačítko zobrazí podokno Event Properties pro označený řádek (= událost)
- Find – vyhledává událost podle klíčového slova
- Jump to Object – otevře objekt, kterého se událost týká (záznam v Editoru registru, složka v Průzkumníku souborů apod.)

- Show Registry Activity – rychlé filtrování zaznamenaných událostí týkajících se registrů ve výpisu
- Show File System Activity – rychlé filtrování událostí souvisejících se souborovým systémem
- Show Network Activity – rychlé filtrování událostí síťového provozu
- Show Process and Thread Activity – rychlé filtrování událostí týkajících se aktivit na úrovni procesů a vláken
- Show Profiling Events – rychlé filtrování událostí zachycujících spotřebu systémových prostředků procesy

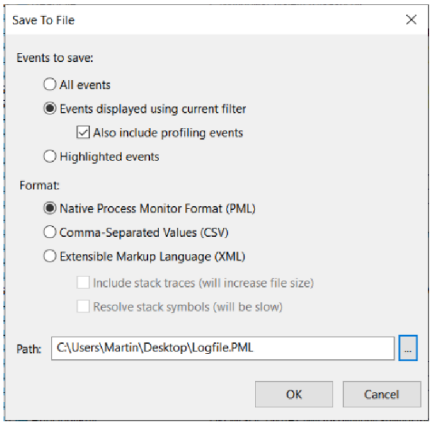
3. Lišta pro menu

a. File

Jako první zde najdeme záložku File, která nám umožní následující operace:

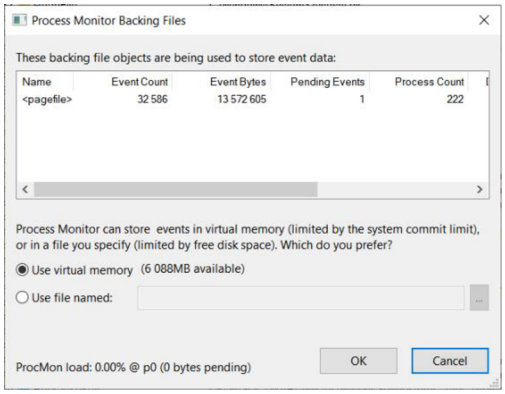
- Uložit kompletní nebo filtrovaný výpis událostí ve formátu .pml, .xml nebo .csv

- Načíst dříve vytvořený výpis událostí, který byl uložen ve formátu .pml (jiný formát není podporován)



Obrázek 34 – Možnosti uložení výpisu

- Nastavení zálohování, tedy kam má Process Monitor ukládat informace o událostech – na výběr je virtuální paměť (velikost omezena nastavením systému) nebo soubor na disku (omezeno jen dostupným volným místem)



Obrázek 35 – Nastavení místa pro výpisy událostí

- Zapínání a vypínání záznamu událostí (totéž jako u tlačítka Capture)

- Export a import konfigurací – zde se myslí hlavně předem vytvořené filtry (Selecký [4 – str.117] podotýká, že toto je využitelné např. častém střídání jednotlivých nastavení monitorování nebo při sdílení nastavení s kolegy)

b. Edit

V této záložce najdeme jednoduché příkazy pro nalezení konkrétní události podle klíčových slov, zkopírování výpisu aktuálně vybrané události, povolení automatického scrollování během záchytu a výmaz zachycených událostí (totéž jako tlačítko Clear).

c. Event

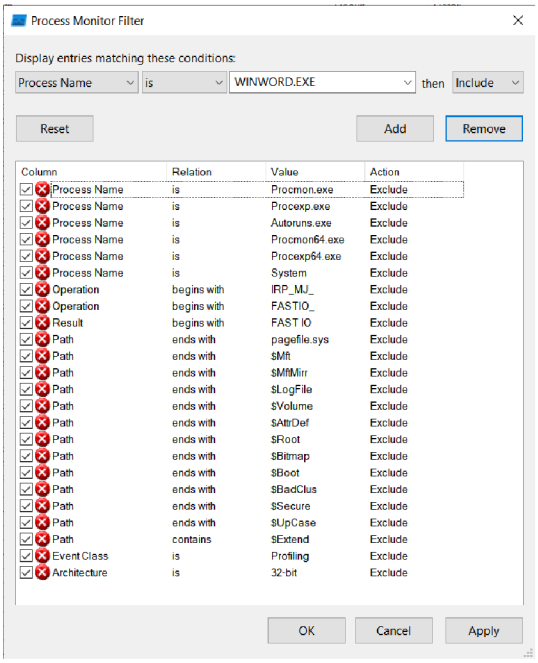
Záložka Event je totožná s nabídkou po kliknutí pravým tlačítkem do okna programu (viz strana 26, obr. 32).

d. Filter

V záložce Filter máme možnost nastavit výstup programu podle svých preferencí. Tlačítky na liště jsme měli možnost ovlivňovat pouze zobrazení jednotlivých událostí podle příslušnosti k jednotlivým třídám, zde je možno nastavit mnohem přesnější filtry.

Jako první položku v menu najdeme Enable Advanced Output, což z implicitních filtrů odstraní ty, které nám neumožňují zobrazit události spojené s procesem System, s procesy nástrojů Sysinternals (Process Monitor, Explorer, Autoruns) a události spojené s některými operacemi a cestami v systému. Zaškrtnutím této možnosti tak zůstane aktivní jen filtr pro události spojené s profilováním procesů a všechny uživatelem ručně vytvořené filtry. Nevýhodou je, že množství vypisovaných událostí se řádově zvýší, jejich zpracování je tak pomalejší a méně přehledné.

Po kliknutí na položku Filter se zobrazí podokno pro správu filtrů. Zde vidíme aktivní filtry, pomocí nabídky v horní části okna pak můžeme vytvářet nové nebo některé z aktuální konfigurace odstranit. Tvorba nových filtrů je založena na výrokové logice. Vždy nejprve volíme kategorii, podle které chceme třídit (název procesu, jeho id, popis, id vlákna, čas a popř. datum apod.). Následuje podmínka (rovná/nerovná se, větší/menší než, začíná/končí s, obsahuje), pak samotná hodnota pro danou kategorii (pokud jsme zvolili název procesu, pak můžeme vybírat z názvů všech procesů zachycených ve výpisu apod.) a nakonec zda má tento filtr odpovídající události do výpisu zahrnout, nebo je naopak skrýt. Tlačítkem Add přidáme nově vytvořený filtr do seznamu aktivních. Pokud si v tomto seznamu některý filtr označíme, můžeme ho kliknutím na tlačítko Remove odebrat. Filtry na aktuální výpis aplikujeme tlačítkem Apply. A tlačítko Reset slouží k obnovení původního nastavení filtrů v programu (viz obrázek 36).



Obrázek 36 – Podokno pro správu filtrů s implicitními filtry

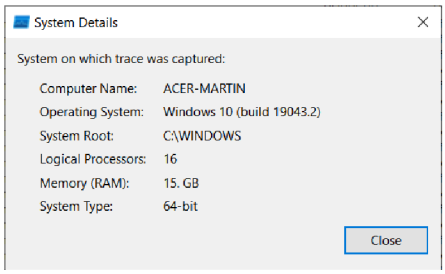
Položkami Save/Load Filter můžeme ukládat aktuální nastavení do seznamu konfigurací filtrů. Tyto pak můžeme používat opakovaně a není třeba je pokaždé manuálně vytvářet.

Doposud jsme aplikací filtrů jednotlivé události pouze skrývali, jejich změnou se tedy změnil i samotný výpis. Což ale znamená, že i události, které pro nás nemají význam, využívají místo v paměti. Tomu můžeme zabránit zaškrtnutím položky Drop Filtered Events. Jak poznamenává Selecký [4–str.121], filtrování se změní v destruktivní a neodpovídající záznamy jsou automaticky zahozeny. Po změně filtrů tak nedojde ke změně výpisu, protože nemá potřebné informace.

A konečně tlačítkem Highlight vyvoláme podokno, kde stejnou logikou, jako při tvorbě filtrů vytvoříme pravidla pro zvýraznění odpovídajících událostí ve výpisu (implicitně jde o podbarvení řádků azurovou barvou). Tlačítkem Make Filter pak pravidla pro zvýrazňování událostí zkopírujeme do dříve popsaných filtrů.

e. Tools

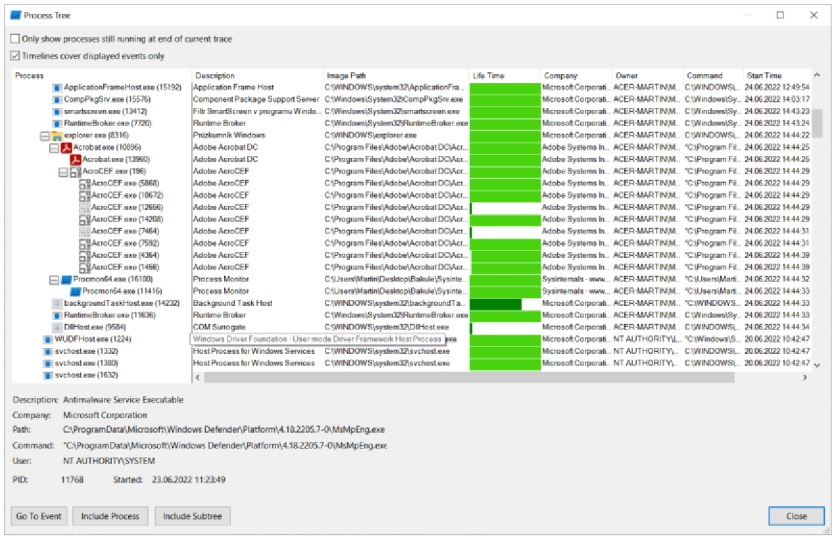
V záložce Tools najdeme sumarizační nástroje, které statisticky zpracovávají získaná data a poskytují nám o nich grafický přehled.



Obrázek 37 – Tools – System Details

První položkou System Details můžeme otevřít okno se stručnými informacemi o systému, na kterém byl daný výstup pořízen (což se hodí pro orientaci v případě analýzy výstupů z vícero počítačů).

Dále máme položku Process Tree, která nám otevře podokno se seznamem procesů, které se svými operacemi podílely na zaznamenaných událostech. Procesy jsou zobrazeny ve stromovém uspořádání, podobně jako u Process Exploreru (viz strana 7).



Obrázek 38 – Tools – Process Tree

Ke každému z nich pak máme detailní informace, jako je jejich popis, umístění souboru, čas spuštění/ukončení nebo lištu, na které je graficky vykreslen čas, po který byl daný proces spuštěn ve vztahu k časové ose zaznamenaných událostí. Ve spodní části máme tlačítka Go To Event, které nám ve výpisu označí první událost vyvolanou daným procesem a tlačítka Include Process / Include Subtree, které slouží k rychlému vytvoření filtrů, které nám opět zajistí, že ve výpisu zůstanou jen události s daným(-i) procesem(-y) souvisí.

Dalších 6 položek nám zobrazí různá podokna poskytující výše zmíněný statistický přehled o zachycených datech:

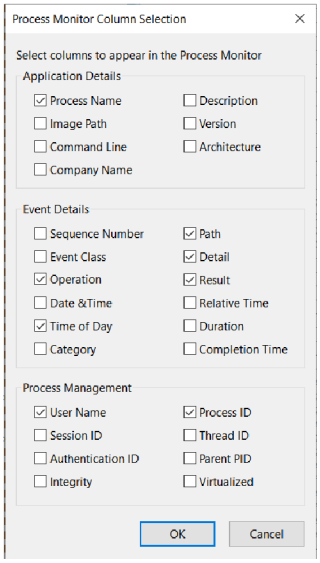
- Process Activity Summary – poskytne výpis všech procesů, podobně jako Process Tree (ale bez hierarchické struktury) a ke každému z nich sadu minigrafů o jejich aktivitě v záznamu (využití CPU, paměti, sítě a podobně v čase)
- File Summary – shrnuje informace o přístupech k jednotlivým souborům v zaznamenaných událostech
- Registry Summary – podobně jako u souborového shrnutí, sumarizuje informace o přístupech k jednotlivým záznamům v registru systému
- Stack Summary – zde najdeme shrnutí operací vyskytujících se v zásobnících volání jednotlivých procesů (viz str. 25), zobrazené ve stromové struktuře
- Network Summary – další shrnutí týkající se tentokrát síťové aktivity, rozdělené podle adres spojení
- Cross Reference Summary – a toto poslední shrnutí zobrazuje cesty k souborům v systému, které byly využity více než jedním procesem (např. v situaci, kdy jeden proces do souboru provedl zápis a další z něj naopak četl)

A konečně položka Count Occurences zobrazuje, jak je patrné už z názvu, jednoduchý výpis počtu událostí vztahujících se k dané zvolené proměnné. Kdy např. jako proměnnou zvolíme jméno procesu, ve výpisu budou názvy jednotlivých procesů a ke každému počet událostí, které k němu byly zaznamenány.

f. Options

Tato záložka poskytuje nastavení programu. Jde například o možnost, že okno Process Monitoru bude neustále v popředí, dále jednotlivé barvy, fonty (včetně velikosti písma) a styl programu (vedle standardního, který je viditelný na obrázcích, máme k dispozici také tmavý mód).

Vedle těchto grafických nastavení jsou zde také konfigurovatelné možnosti výpisu. Položkou Select Columns můžeme ovlivnit, které informace k jednotlivým událostem bude hlavní okno zobrazovat. Implicitně zaškrtnuté sloupce lze vidět na obrázku 39 (a daný výpis pak na obrázku 28 – str. 23).



Obrázek 39 – Options – Select Columns

Dále je zde možnost pomocí položky History Depth nastavit omezení počtu zaznamenaných událostí (buď omezením místa, které mohou události zabrat, nebo času, po který budou zaznamenávány) a také možnost nastavit tzv. Ring Buffer. Process Monitor pak bude nejnověji zaznamenanými událostmi přepisovat ty nejstarší (chovat se tedy podobně jako například autokamera).

A konečně je třeba zmínit položku Enable Boot Logging, která nám umožní zaznamenat události probíhající během startu operačního systému. Toto je podrobněji předvedeno v praktické části, v úloze 4.4 – Analýza spouštění PC.

Poznámka: Při Boot Loggingu nejsou k dispozici záznamy, pro jejichž zachycení je třeba služby ETW (např. síťové události –str. 23).

g. Help

Poslední záložkou je obligátní nápověda, ze které si můžeme otevřít soubor kompilované nápovědy, distribuovaný v balíku Sysinternals Suite nebo otevřít podokno zobrazující argumenty pro spuštění a ovládání Process Monitoru z příkazové řádky (obr. 40).

Obrázek 40 – Argumenty pro příkazovou řádku

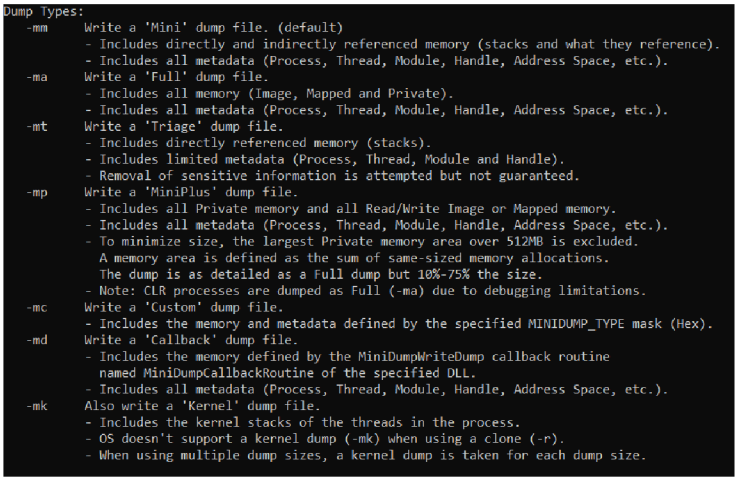
3.1.3 ProcDump

ProcDump je, na rozdíl od dvou výše popsaných, aplikací pro konzolový řádek. Byla vytvořena roku 2009 autory Markem Russinovichem a Andrewem Richardsem [1-str.194]. Umožňuje nám monitorovat běh procesu (dle zadaných parametrů již běžícímu nebo takovému, který bude teprve spuštěn) a generovat výpisy paměti daného procesu (soubory .dmp – stejný typ souboru – výpisu paměti procesu jako nám umožňuje získat Process Explorer, viz str. 11), například v momentě dosažení předem nastavených podmínek, v případě, kdy dojde v programu k výjimce, když program přestane odpovídat nebo třeba s předem pevně danou frekvencí, nezávisle na okolnostech. Nástroj lze též nastavit jako automatický debugger (AeDebug) v systému Windows, v tom případě je automaticky spuštěn při každém pádu procesu a zachytí okolnosti, které k němu vedly. Praktickou ukázku použití najdete v příkladu 4.5.

Poznámka: Jak se můžeme dočíst v dokumentaci [5], existuje také (poměrně nová) verze nástroje ProcDump pro operační systémy Linux Fedora 29, Ubuntu 16.04 LTS a CentOS 7 / Red Hat Enterprise Linux.

Tyto výpisy mohou být užitečné hlavně při testování vyvíjených aplikací, protože případné chyby v kódu (například neošetřené výjimky) se nemusí projevovat jen pády aplikace, ale samozřejmě také pomalejším během aplikace nebo abnormálně vysokým využitím operační paměti nebo procesoru počítače.

V syntaxi příkazů pro používání nástroje ProcDump určujeme typ .dmp souboru, podmínky, při kterých má dojít k jejich vytvoření, proces, který bude monitorován nebo také to, kam se budou tyto výpisy paměti ukládat.



Obrázek 41 – Výpis argumentů pro určení typu .dmp souboru

Nejprve je třeba vysvětlit rozdíly mezi jednotlivými typy dump souborů (výpisů paměti, chcete-li). Jejich velikost a zachycené detaily se liší dle použitých argumentů (viz obr. 41). Některé zde popíšeme:

- **Mini dump**
  - Implicitní typ výpisu, který se vytváří buď automaticky (když není použit žádný z argumentů), nebo s argumentem **-mm**.
  - Obsahuje pouze základní informace o procesu a jeho vláknech (včetně záznamů o využití procesoru jednotlivými vlákny) - záznam PEB (Process Enviroment Block) a TEB (Thread Enviroment Block), výpis ze zásobníku volání, výpis registrů, seznam použitých modulů (viz str. 14) a souborů používaných procesem (objekty handle - str. 8), popis virtuálního adresního prostoru paměti a některé části z paměti využívané tímto procesem

- **Mini Plus dump**
  - Výpis získáme pomocí argumentu **-mp**
  - Tento typ byl vytvořen pro tvorbu výpisů u programů, které používají velké množství operační paměti (typicky serverové aplikace) a u kterých by vytvoření plného výpisu (full dumpu) zabralo neúměrné množství času
  - Pomocí heuristiky si určí, které části paměti si má navíc přidat ke standardnímu Mini dumpu tak, aby byl debugging pomocí Mini Plus výpisu stejně efektivní jako v případě plného výpisu

- **Full dump**
  - Vznikne při použití argumentu **-ma**
  - Obsahuje totéž, co mini dump a navíc kopii uživatelského virtuálního adresního prostoru použitého procesem (Image, Stack, Heap, Data, Code segmenty [2 – str.525-528])
  - Největší, nejpodrobnější, nejdéle vytvářený

- **Custom dump**
  - Pomocí argumentu **-mc** si pokročilý vývojář může vytvořit vlastní pravidla pro vytvoření výpisu paměti, tedy sám si určit, co bude zaznamenáno
  - Použije k tomu rozhraní MiniDumpApiSet.h, vytvoří vlastní .dll knihovnu (viz str. 7), v ní přepíše funkci MINIDUMP_CALLBACK_ROUTINE [1-str.210; 6]
  - Při spuštění ProcDumpu pak musí uživatel specifikovat cestu k této knihovně

Poznámka: Při vytváření jakéhokoli typu výpisu je daný proces pozastaven (SUSPENDED – str. 8). Na toto je třeba brát zřetel např. při ladění serverového softwaru v produkční fázi, kdy může být problematické zachycení full dumpu procesu využívajícího desítky GB RAM paměti daného stroje. Nejen že samotný výpis paměti bude mít opět desítky GB, ale po dobu jeho vytváření nebude tento proces poskytovat své služby a server tedy bude nedostupný. Toto lze vyřešit pomocí argumentu ***– r****, čtěte dále.*

Nyní je třeba určit podmínky, za kterých dojde k zachycení výpisu (vytvoření dump file), jde například o:

- **-c/-cl**: vytvoření daného typu výpisu při dosažení nastavené úrovně vytížení CPU (nebo při poklesu pod danou úroveň pro – cl)
- **-m/-ml**: nastavení podobné podmínka jako u argumentu – c, jen pro operační paměť využívanou procesem
- **-n**: určení počtu výpisů, které mají být při provedení daného příkazu zachyceny

```
Conditions:
  -a      Avoid outage. Requires -r. If the trigger will cause the target
          to suspend for a prolonged time due to an exceeded concurrent
          dump limit, the trigger will be skipped.
  -at     Avoid outage at Timeout. Cancel the trigger's collection at N seconds.
  -b      Treat debug breakpoints as exceptions (otherwise ignore them).
  -c      CPU threshold above which to create a dump of the process.
  -cl     CPU threshold below which to create a dump of the process.
  -dc     Add the specified string to the generated Dump Comment.
  -e      Write a dump when the process encounters an unhandled exception.
          Include the 1 to create dump on first chance exceptions.
  -f      Filter (include) on the content of exceptions and debug logging.
          Wildcards (*) are supported.
  -fx     Filter (exclude) on the content of exceptions and debug logging.
          Wildcards (*) are supported.
  -g      Run as a native debugger in a managed process (no interop).
  -h      Write dump if process has a hung window (does not respond to
          window messages for at least 5 seconds).
  -k      Kill the process after cloning (-r), or at end of dump collection.
  -l      Display the debug logging of the process.
  -m      Memory commit threshold in MB at which to create a dump.
  -ml     Trigger when memory commit drops below specified MB value.
  -n      Number of dumps to write before exiting.
  -o      Overwrite an existing dump file.
  -p      Trigger on the specified performance counter when the threshold
          is exceeded. Note: to specify a process counter when there are
          multiple instances of the process running, use the process ID
          with the following syntax: "\Process(<name>_<pid>)\counter"
  -pl     Trigger when performance counter falls below the specified value.
  -r      Dump using a clone. Concurrent limit is optional (default 1, max 5).
          OS doesn't support a kernel dump (-mk) when using a clone (-r).
          CAUTION: a high concurrency value may impact system performance.
          - Windows 7    : Uses Reflection. OS doesn't support -e.
          - Windows 8.0 : Uses Reflection. OS doesn't support -e.
          - Windows 8.1+: Uses PSS. All trigger types are supported.
  -s      Consecutive seconds before dump is written (default is 10).
  -t      Write a dump when the process terminates.
  -u      Treat CPU usage relative to a single core (used with -c).
  -w      Wait for the specified process to launch if it's not running.
  -wer    Queue the (largest) dump to Windows Error Reporting.
  -x      Launch the specified image with optional arguments.
          If it is a Store Application or Package, ProcDump will start
          on the next activation (only).
  -64     By default ProcDump will capture a 32-bit dump of a 32-bit process
          when running on 64-bit Windows. This option overrides to create a
          64-bit dump. Only use for WOW64 subsystem debugging.
```

Obrázek 42 – Výpis parametrů pro nastavení monitorování procesu

- **-e**: zachycení a tvorba výpisu v případě, že v programu dojde k neošetřené výjimce

- **-t**: vytvoření výpisu při ukončení programu

- **-i**: instalace ProcDumpu jako AeDebuggeru – tedy kdykoli v systému Windows dojde např. k pádu aplikace, je spuštěn ProcDump, který zachytí do výpisu stav procesu, který pádu těsně předcházel -> můžeme pak provést tzv. Post-Mortem Debugging, ladění programu za situace, kdy daný program již neběží

- **-r**: před začátkem vytváření výpisu je daný proces (v daném stavu) zkopírován, tato kopie je následně pozastavena a jsou z ní získány informace pro dump file -> původní proces tak může pokračovat v běhu a poskytovat své služby, i když zachycení výpisu kopie stále probíhá (v ukázce 4.3 na obrázcích 69 a 70 je vidět, že záchyt i relativně malých výpisů trvá cca několik sekund)

Nakonec si můžeme určit jméno vytvářeného výpisu a místo, kam má být uložen, o tom více v praktické ukázce 4.3.

3.2 Sysinternals Security Utilities

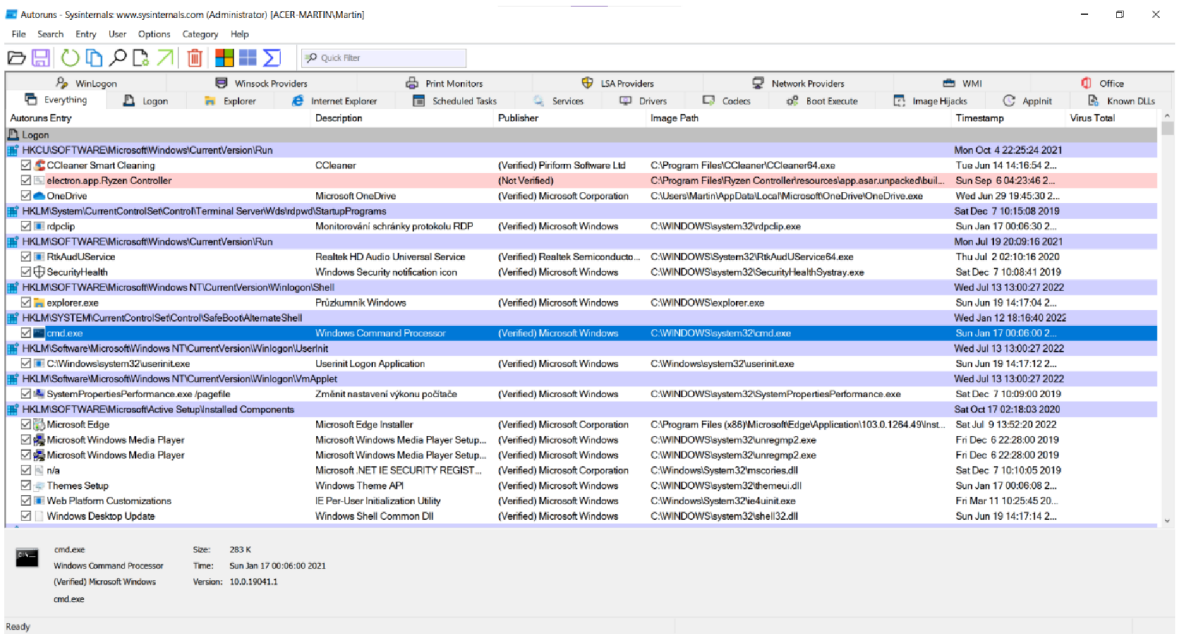
V této části si popíšeme některé stále využitelné bezpečnostní nástroje ze skupiny Sysinternals. V roce 2005 Mark Russinovich na svém blogu (článek lze stále nalézt v archivu [7]) uveřejnil článek o rootkitech využívaných společností Sony BMG při používání jimi vydaných originálních CD nosičů. Tyto se automaticky a bez vědomí uživatele instalovaly při prvním vložení disku do počítače (je nutné si uvědomit, že tyto CD disky k přehrávání svého obsahu používaly proprietární přehrávače). Důležité je, že Mark k tomuto odhalení použil jeden ze svých nástrojů, RootkitRevealer. Vzhledem k nekompatibilitě s operačními systémy založenými na 64bitové architektuře již není RootkitRevealer součástí distribuovaného balíku Sysinternals Suite. Popíšeme si tak některé další nástroje, které jeho součástí jsou – **Autoruns** a **System Monitor**.

3.2.1 Autoruns

Nástroj Autoruns stojí na pomezí mezi nástroji procesními (Process Utilities dle Russinoviche [1]) a bezpečnostními (Security Utilities dle Seleckého [4]). Slouží nejen k zobrazení, ale také ke správě aktuálně nastavených, automaticky se spouštějících aplikací. Nejde pouze o programy, které se spouští po startu systému (jako u záložky Po spuštění, kterou lze nalézt v Task Manageru – Správci úloh operačního systému Windows), ale také o služby systému Windows, ovladače, plánované úlohy a podobně. Podobně jako u Process Exploreru, lze z nástroje Autoruns měnit tuto konfiguraci. Je možno dočasně zakázat spouštění daného programu (prostým odškrtnutím u daného záznamu), nebo ho z konfigurace i vymazat. Tento krok je ale nevratný, je třeba ho vždy potvrdit v dialogovém okně (uživatel tedy musí vždy vědět, co maže). Pro vyzkoušení tohoto nástroje tedy autor této práce důrazně doporučuje použít virtuální stroj (viz praktická úloha 1).

V balíku si uživatel vybere verzi podle architektury svého operačního systému (32 nebo 64bitovou). Po spuštění programu se otevře hlavní okno a dojde k výpisu mnoha různých položek.

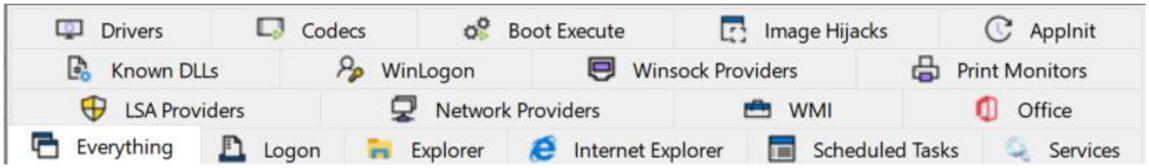
Hlavní okno programu vypadá takto:



Obrázek 43 – Hlavní okno Autoruns

Po spuštění programu dojde nejprve k identifikaci tzv. ASEP -> Autostart Extensibility Point (na obrázku 43 vyznačeny jako světle fialové řádky). Jde o místa v operačním systému Windows, která jsou konfigurována tak, aby automaticky spouštěla programy (vykonávala kód v nich obsažených) při dosažení určitých podmínek, jako je start systému, načtení vybraných ovladačů a knihoven, dosažení času (např. vypínání systému Windows 10 z důvodu instalace automatických aktualizací po ukončení doby nastavené uživatelem), připojení uživatele k internetu a podobně. Tyto body jsou ale často zneužívány tvůrci malwaru ke spouštění škodlivého kódu [8 – část A taxonomy of Windows auto-start extensibility points], mohou být využívány bloatwarem předinstalovaným výrobcem počítače nebo omylem nainstalovaným v průběhu instalace jiného programu.

Jednotlivé ASEP jsou následně rozděleny do 18 různých kategorií (na obrázku 43 je lze vidět jako záložky mezi ikonami tlačítek a výpisem), mezi kterými může uživatel přepínat, nebo si otevřením 19. karty Everything zobrazit všechny body najednou.



Obrázek 44 – Karty rozdělení ASEP

Pozn. Kniha Troubleshooting with the Windows Sysinternals Tools [1] Marka Russinoviche na straně 124 popisuje 19 různých kategorií, nicméně novější verze nástroje Autoruns nezobrazují kartu Sidebar Gadgets (pro miniaplikace na postranním panelu, běžné ve Windows Vista a 7, které se v novějších systémech již nevyskytují).

Z 18 kategorií ASEP lze vyzdvihnout například následující:

- **Logon** – zahrnuje taková umístění, která jsou využívána programy pro spuštění po přihlášení uživatele – je tedy nejpodobnější dříve zmíněné záložce Po spuštění, kterou najdeme v okně Správce úloh

- **Drivers** – v této záložce jsou uvedeny ovladače, které systém používá – ovladače (stejně jako služby) využívají ASEP s adresou **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services**

  Pozn.: V záznamech na kartě Drivers lze nalézt nejen detaily ovladačů, které lze zachytit Protokolem spouštění (Boot Log), který si uživatel s dostatečným oprávněním může nechat vygenerovat po startu systému (volbu je nutné zaškrtnout v okně Konfigurace systému – msconfig), ale také mnoho dalších

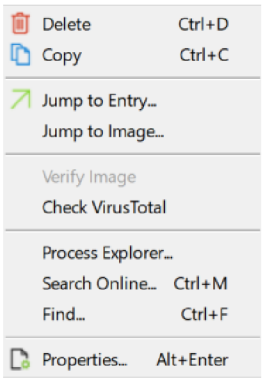
- **Services** – zde jsou uvedené služby, které systém Windows používá a které nejsou zablokovány (služby pozastavené nebo zablokované z programu zobrazeny jsou a v registru je jim přiřazena hodnota AutorunsDisabled)

- **Scheduled Tasks** – zde vypsané položky jsou nástroje programů, které po spuštění systému Windows čekají ve stavu Připraveno na aktivační úlohu (po přihlášení konkrétního uživatele, v konkrétní čas, po uplynutí nastavené doby) -> při otevření umístění dané položky dojde k otevření okna Plánovač úloh

- **Image Hijacks** – položky na této kartě jsou podezřelé z toho, že se maskují za jiný program v PC (v čistém počítači je tato záložka prázdná, nanejvýš zde můžeme vidět Process Explorer spouštěný místo Správce úloh – viz str.17)

Následně jsou k jednotlivým bodům vypsány soubory a programy, které je využívají. Autostart Extension Points, které v systému nejsou žádným programem využívány, jsou implicitně skryté, lze si je ale zobrazit příslušnou položkou v menu (viz dále).
Pro každý záznam jsou zobrazena následující data:

- Jméno záznamu (program, ovladač, služba, …)
- Jeho krátký slovní popis
- Jméno vydavatele a ověření (tedy potvrzení o pravosti digitálního podpisu)
- Image Path, tedy cesta ke spustitelnému souboru, knihovně nebo ovladači (soubor s příponou .sys)
- Časové razítko (Timestamp)
- Ověření v databázi VirusTotal (viz Process Explorer, str. 10) – implicitně je sloupec prázdný, protože volba pro skenování záznamů v databázi VirusTotal.com není aktivní (skenování všech záznamů je časově náročnější a zpomalovalo by spouštění programu) -> toto lze aktivovat v menu (viz dále); ověření jednoho záznamu lze provést z rychlé nabídky (viz další strana)
    - Samozřejmou podmínkou správného fungování této funkce je aktivní připojení k internetu



Obrázek 45 – Rychlé možnosti

Po kliknutí pravým tlačítkem na záznam se zobrazí rychlá nabídka, podobná jako u Process Monitoru. Položkou **Delete** dojde k výše zmíněnému nevratnému odebrání daného programu z ASEP, a tedy ke znemožnění jeho automatického spouštění. Pomocí **Copy** zkopírujete do schránky detaily z daného záznamu ve formě prostého textu. Položka **Jump to Entry** otevírá Editor registru systému Windows ve složce odpovídající danému ASEP, zatímco **Jump to Image** otevírá složku souborového systému, ve které je umístěn spouštěcí soubor pro daný záznam. Pomocí **Check VirusTotal** provedete dotaz na webovou službu VirusTotal.com a získáte skóre pro daný záznam (program). Položkou **Process Explorer** otevřete stejnojmenný nástroj, **Search Online*** vyhledá název programu ze záznamu online. **Find** pomůže vyhledat záznam

podle klíčového slova a **Properties** otevře standardní okno Vlastnosti pro daný spustitelný soubor.

**Pozn.: Autor této práce zjistil, že ne všechny prohlížeče internetu, nastavené jako výchozí v daném systému jsou funkční. Bez problémů funkční je Microsoft Edge, naopak s nastavenou Mozillou Firefox k žádnému vyhledání nedojde (Autoruns ve verzi 14.05).*

Podobně jako u ostatních nástrojů Sysinternals s grafickým výstupem je k dispozici lišta s tlačítky pro rychlou volbu některých funkcí (některé jsou totožná s rychlou volbou pravým tlačítkem a všechny jdou také najít v menu).

Obrázek 46 – Tlačítka pro rychlou volbu

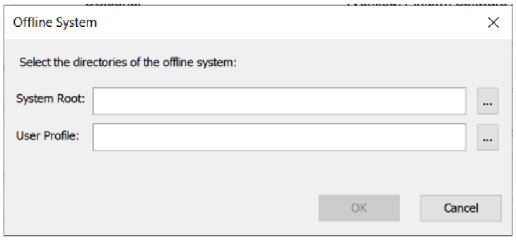
Zleva jde o následující funkce:

- **Open**, **Save** – podobně jako dříve zmiňované nástroje Sysinternals i Autoruns umožňuje otevření dříve zachycené nebo uložení aktuální konfigurace automaticky spouštěných programů pro pozdější analýzu (používá proprietární formát .arn)
- **Refresh** – obnovení záznamů
- **Copy** – zkopírování vybraných záznamů do schránky
- **Find** – otevře dialogové okno pro vyhledávání
- **Properties** – otevření okna Vlastnosti (viz předchozí strana)
- **Jump** – odpovídá Jump to Entry (viz předchozí strana)
- **Clear** – opět možnost smazání (viz předchozí strana)
- **Hide Microsoft Entries*** – volbou této možnosti skryjeme záznamy pro programy podepsané firmou Microsoft
- **Hide Windows Entries** – totéž, ale týká se pouze ověřených částí operačního systému (například většina služeb, automatické aktualizace apod.)
- **Hide VirusTotal Clean Entries** – skryje záznamy se skóre 0 (ve formátu 0/N, kde N je počet databází antivirových programů v databázi VirusTotal), nicméně až po provedení skenování (viz dále)

Pozn.: Můžeme si zvolit možnost skrýt položky společnosti Microsoft, ale jak se lze dočíst v knize Windows Internals Part II [3 – str. 837], je důrazně doporučeno toto spojit s možností **Verify Image Signatures (tedy ověření digitálního podpisu programu), jinak by mohlo dojít ke skrytí záznamu o malwaru, který se jako program od společnosti Microsoft pouze maskuje.*

V menu programu lze nalézt následující položky:

- **File**
  - Uložení / otevření konfigurace
  - Analýza konfigurace u offline systému (např. při použití live USB a načtení nového OS získáme přístup k souborovému systému OS počítače – pak do řádku System Root zvolíme složku Windows a jako User Profile najdeme cestu ke složce uživatele, např. Users/Martin)

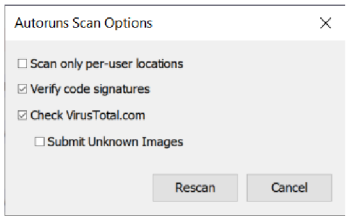


Obrázek 47 – Analýza offline systému

  - Porovnání aktuální konfigurace s některou dříve uloženou -> pokud se různí, jsou tyto rozdíly vypsány (červené podbarvení = chybí, zelené = jsou navíc)

- **Search**
  - Obsahuje několik možností pro vyhledávání ve výpisu
- **Entry**
  - Obsahuje položky totožné s těmi v rychlé nabídce (po kliknutí pravým tlačítkem na záznam – str.40)
- **User**
  - Tato menu nabídka se zobrazí jen tehdy, je-li Autoruns spuštěn se zvýšeným oprávněním (volba Spustit jako správce)
  - Umožňuje zkoumat autostart konfigurace jednotlivých uživatelů, kteří se přihlásili k danému PC – po zvolení uživatele ze seznamu Autoruns znovu provede sken systému (včetně ověření vstupů proti databázi VirusTotal, byla-li tato možnost povolena)

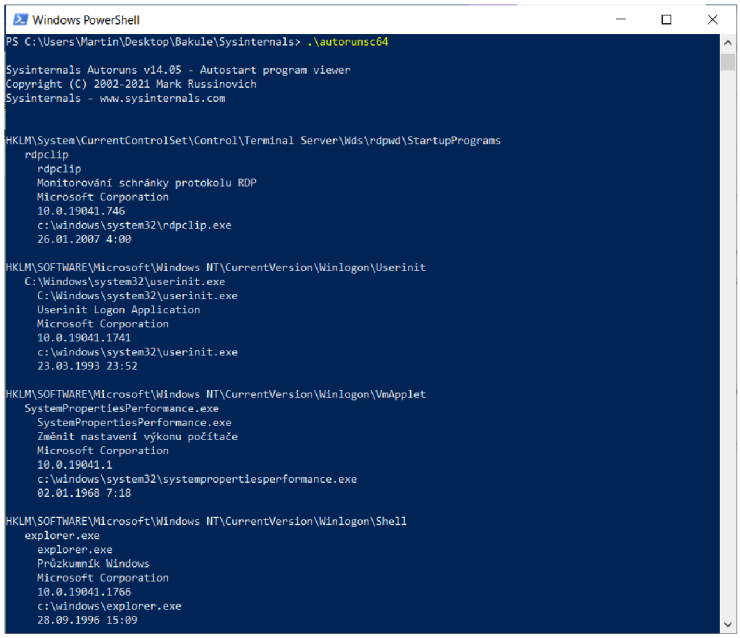
- **Options**
  - První položkou je implicitně aktivní možnost **Hide Empty Locations**, která skryje ASEP, která aktuálně nejsou využívána
  - Dále máme položky Hide Microsoft, Windows a VirusTotal Clean Entries (viz str. 41)
  - A nakonec jsou zde možnosti nastavení skenování (viz obr. 48) a zobrazení GUI programu



Obrázek 48 – Nastavení možností skenování

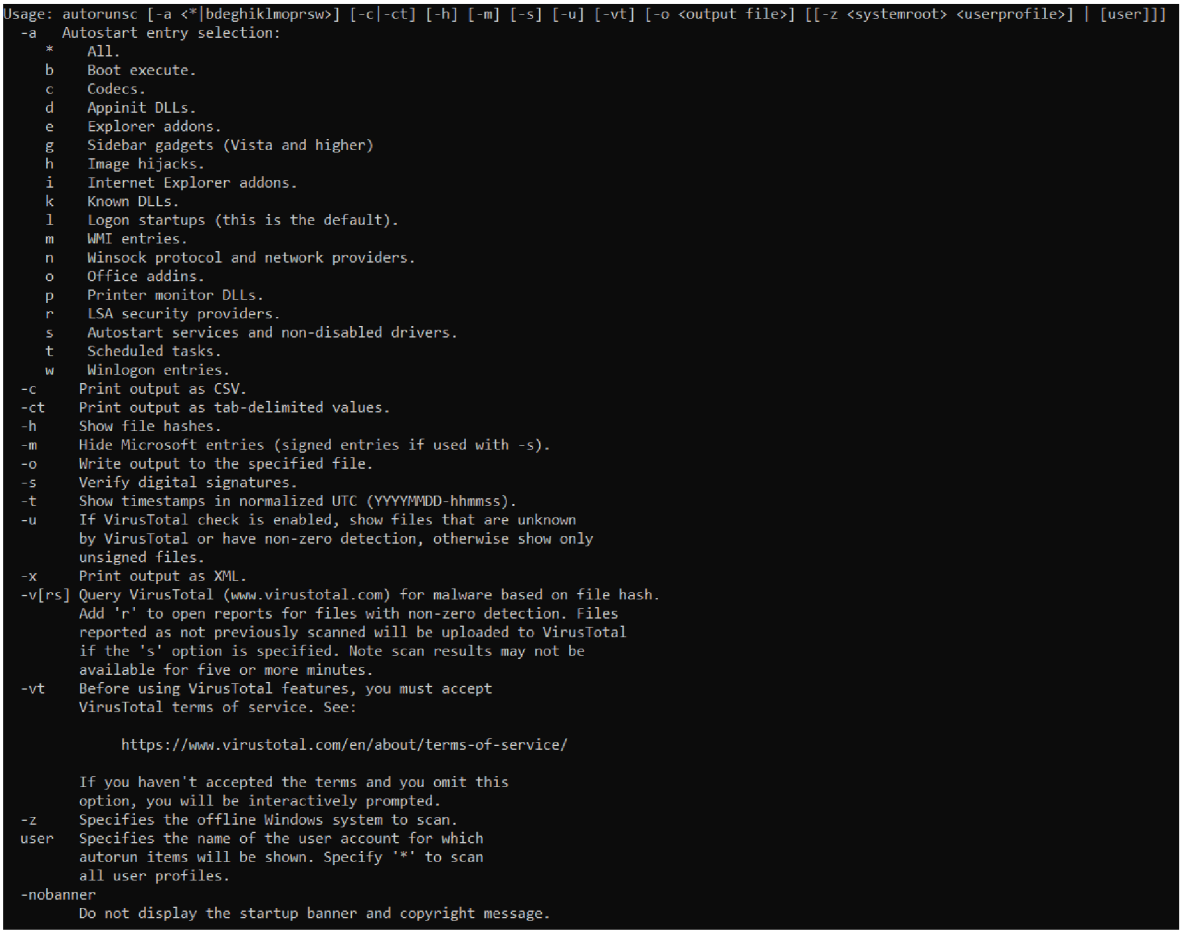
- **Category**
  - Přepínání mezi jednotlivými kategoriemi ASEP, totožné jako rychlé přepínání pomocí karet (viz obr. 44 na straně 39)
- **Help**
  - Možnost otevření webové stránky nástroje Autoruns v rámci Microsoft Docs (totožné se zdrojem č. 9) a standardního okna se základními informacemi o programu

Jak se můžeme dočíst v dokumentaci k nástroji Autoruns [9], máme k dispozici také verzi s rozhraním pro příkazový řádek – AutorunsC/AutorunsC64. Po zadání příkazu autorunsc64 dojde k výpisu podrobností ke každému identifikovanému ASEP do příkazové řádky (příklad na obrázku 44). Tuto verzi lze použít pro automatizovanou správu systému pomocí skriptů např. v prostředí Powershell, umožňuje také snadný export dat ve formátu .csv nebo .xml.



Obrázek 49 – Ukázka Autorunsc64 pro Powershell

Kompletní výpis příkazů pro ovládání lze nalézt ve výše zmíněné dokumentaci nebo po zadání argumentu **-?** (obrázek 50).



Obrázek 50 – Nápověda pro ovládání Autorunsc

3.2.2 System Monitor

System Monitor (nebo též SysMon) je CLI aplikace pro dlouhodobé sledování počítače. Po instalaci je spouštěn vždy při startu systému, běží na pozadí, monitoruje běh počítače a zachytává události podle předem nastavené konfigurace. Takto zachycené události si lze prohlédnout v Prohlížeči událostí (Windows Event Log) a zobrazit si podrobnosti. Tímto monitorováním (které navíc začíná brzy po startu počítače během načítání operačního systému) lze odhalit nežádoucí chování softwaru nebo přímo škodlivý kód.

Během instalace Sysmon je do počítače nainstalován příslušný ovladač (Driver) běžící v kernel módu a také služba (Service), běžící v uživatelském módu. Právě ta je zodpovědná za filtrování událostí a také hlídá změny v nastavení aplikace v případě, že uživatel změnil konfigurační soubor (nebo nahrál jiný). Dále můžeme zvolit soubor s konfigurací filtrování, ten si musíme připravit předem (ve formátu .xml). Pokud nenastavíme cestu k tomuto souboru, bude Sysmon monitorovat pouze implicitně nastavenou tvorbu a ukončování procesů. Ukázku konfiguračního souboru lze nalézt v praktické úloze 4.5. V době psaní této práce Sysmon podporoval 26 různých typů událostí (plus jeden představující chybu programu), které lze zachytit (všechny si lze prohlédnout v dokumentaci k programu [10]). Jde například o:

- Vytvoření, ukončení nebo manipulace s běžícím procesem
- Operace se souborovým systémem – tvorba, smazání, změna časového razítka souboru
- Manipulace s registry systému Windows – vytvoření nového objektu, zadání nové hodnoty již existujícímu apod.
- Některé síťové aktivity

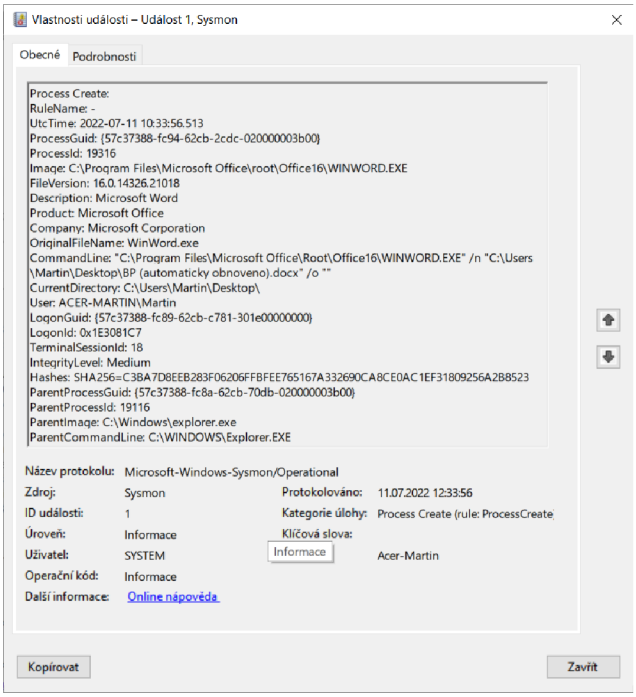
Program se ovládá z příkazové řádky, voláme příkazem *sysmon/sysmon64* (dle architektury OS) a přidáme některý ze základních argumentů:

- **-i** – instaluje System Monitor do počítače, můžeme rozšířit o cestu ke konfiguračnímu souboru
- **-u** – odinstaluje program

- **-c** – spolu s cestou k (např. novému) konfiguračnímu souboru aktualizuje nastavení aplikace
- **-s** – vypíše do příkazové řádky obsah aktuálně používaného konfiguračního souboru

Jak bylo uvedeno dříve, vytvořené záznamy můžeme najít v Prohlížeči událostí (Windows Event Log, pokud používáte anglickou verzi OS). Po otevření prohlížeče lze najít události zachycené System Monitorem rozbalením stromu v levé části okna – **Protokoly aplikací a služeb/Microsoft/Windows/Sysmon/Operational.**

V horní části okna se následně zobrazí seznam zaznamenaných událostí od nejnovější po nejstarší (s tím, že ty nejstarší jsou postupně nahrazovány – LIFO). Po dvojím kliknutí na daný řádek události se zobrazí podokno s detailními informace o zachycené události (viz obrázek 43). Tyto informace se liší podle typu zachycené události, pro vytvoření procesu budou informace následující [1-str.324]:



Obrázek 51 – Vlastnosti záznamu události nástrojem System monitor

- čas zachycení události
- ID a GUID* procesu
- Příkaz, kterým byl proces vytvořen (pro příkazovou řádku)
- Cesta ke spustitelnému souboru, který proces vytvořil
- Cesta k adresáři, z kterého byl spuštěn soubor
- Pokud jde o synovský proces, najdeme zde také jméno, id, příkaz příkazové řádky a cestu ke spustitelnému souboru rodičovského procesu

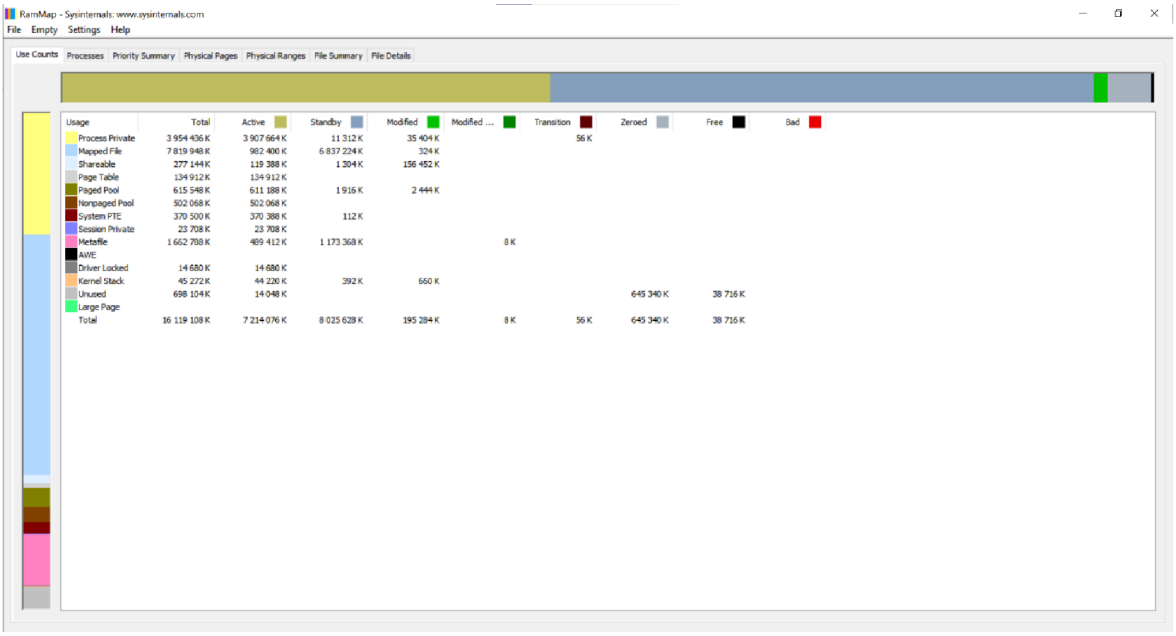
**GUID je unikátní id vytvářené některými nástroji Sysinternals pro identifikaci procesu, jednou přiřazené procesu zůstává i přes restart Sysmonu, vzniká z informací o procesu, je tedy opakovaně znovuvytvořitelný*

3.3 Sysinternals System Information Utilities

Nástroje popsané v této kapitole nám umožní získat ucelené informace o našem i vzdáleném počítači. Tyto informace nejsou zaměřeny na konkrétní proces nebo soubor, ale na systém jako celek a HW zdroje, které nabízí. Najdeme zde jak nástroje pro získání podrobných informací o HW počítače, tak o operačním systému nebo instalovaných aplikacích. Budou zde popsány nástroje **RAMMap**, **PSInfo**, **LoadOrder** a **CoreInfo.**

3.3.1 RAMMap

Tato aplikace nám zobrazuje souhrnné informace o operační paměti v systému Windows [11]. Na rozdíl od informací poskytovaných Process Explorerem (nebo v této práci vynechaným procesním nástrojem VMMap) jde v tomto případě o přehled správcem paměti (memory managerem) alokované fyzické paměti z několika různých pohledů, rozlišených na 7 kartách.



Obrázek 52 – RAMMap – karta Use Counts

- Na obrázku 52 je obsah první karty **Use Counts**, na řádcích vidíme typ alokace paměti (Memory Allocation Type – Process Private, Mapped File, …), sloupce pak představují seznam stránek (Page List – Active, Standby, Zeroed,

Free, …). Nad a vlevo vedle tabulky jsou pak souhrnné grafy představující výše zmíněná rozdělení.

- Druhá karta **Processes**, jak již název napovídá, zobrazuje jednoduchý seznam všech aktivních procesů a přehled jim alokované fyzické paměti

- Na třetí kartě **Priority Summary** najdeme krátký přehled o množství RAM paměti, rozdělené podle priorit (0-7)

- Čtvrtá karta **Physical Pages** zobrazuje detailní informace o každé jednotlivé buňce RAM paměti v počítači. Každý řádek reprezentuje konkrétní buňku, je zde uvedena fyzická adresa v hexadecimálním tvaru (počáteční adresa 0x1000, konečná záleží na množství paměti v PC), do kterého seznamu (Page List) aktuálně patří, její použití (typ alokace – Process Private, Shared, …), proces, který daný úsek paměti využívá a virtuální adresa (opět v hexadecimálním tvaru), kterou se na obsah této stránky dotazuje.

- Ve spodní části okna máme možnost filtrování výstupu, nejprve zvolíme sloupec, podle kterého chceme filtrovat a následně hodnotu, podle které bude výpis omezen

    - Pátá karta **Physical Ranges** nám nabízí přehled o rozsahu fyzických adres

    - Na šesté kartě **File Summary** najdeme cestu ke každému souboru, který je nahrán v RAM a přehled, kolik paměti je mu přiděleno

    - A konečně sedmá karta **File Details** obsahuje detaily k souborům uvedeným v části File Summary

V položce **File** menu můžeme uložit aktuální stav paměti počítače, jak ho zobrazuje RAMMap, do souboru proprietárního formátu .rmp nebo stejný soubot načíst a prohlédnout si již dříve zachycenou situaci.

A nakonec v položce menu **Empty** můžeme vyčistit pracovní sady (Working Set – paměť přidělená procesům) a tabulky stránek paměti. Pro zobrazení změn je třeba aktualizovat výpis klávesou F5 nebo položkou Refresh na kartě File menu.

3.3.2 PsInfo

PsInfo je (jako všechny nástroje skupiny PsTools) aplikace s výstupem na příkazovou řádku, která nám umožňuje získat informace jak o systému, na kterém je spuštěna, tak o vzdáleném počítači v místní síti. Na obrázku 53 můžeme vidět implicitní výstup programu.

```
C:\Users\Martin\Desktop\Bakule\Sysinternals>psinfo

PsInfo v1.78 - Local and remote system information viewer
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

System information for \\ACER-MARTIN:
Uptime:                    Error reading uptime
Kernel version:            Windows 10 Home, Multiprocessor Free
Product type:              Professional
Product version:           6.3
Service pack:              0
Kernel build number:       19043
Registered organization:
Registered owner:          Martin
IE version:                9.0000
System root:               C:\WINDOWS
Processors:                16
Processor speed:           2.8 GHz
Processor type:            AMD Ryzen 7 4800H with Radeon Graphics
Physical memory:           384 MB
Video driver:              NVIDIA GeForce GTX 1650 Ti
```

Obrázek 53 – Implicitní zobrazení PsInfo

Tento výstup lze modifikovat pomocí několika parametrů, jako je například [12]:

- \\computer – jméno vzdáleného počítače v místní doméně, pro který chceme získat výstup

```
Usage: psinfo [-h] [-s] [-d] [-c [-t delimiter]] [filter] [\\computer[,computer[,..]]|@file [-u Username [-p Password]]]
    -u          Specifies optional user name for login to
                remote computer.
    -p          Specifies password for user name.
    -h          Show installed hotfixes.
    -s          Show installed software.
    -d          Show disk volume information.
    -c          Print in CSV format
    -t          The default delimiter for the -c option is a comma,
                but can be overriden with the specified character. Use
                "\t" to specify tab.
    filter      Psinfo will only show data for the field matching the filter.
                e.g. "psinfo service" lists only the service pack field.
    computer    Direct PsInfo to perform the command on the remote
                computer or computers specified. If you omit the computer
                name PsInfo runs the command on the local system,
                and if you specify a wildcard (\\*), PsInfo runs the
                command on all computers in the current domain.
    @file       PsInfo will run against the computers listed in the file
                specified.
    -nobanner   Do not display the startup banner and copyright message.
```

Obrázek 54 – Možné modifikace příkazu

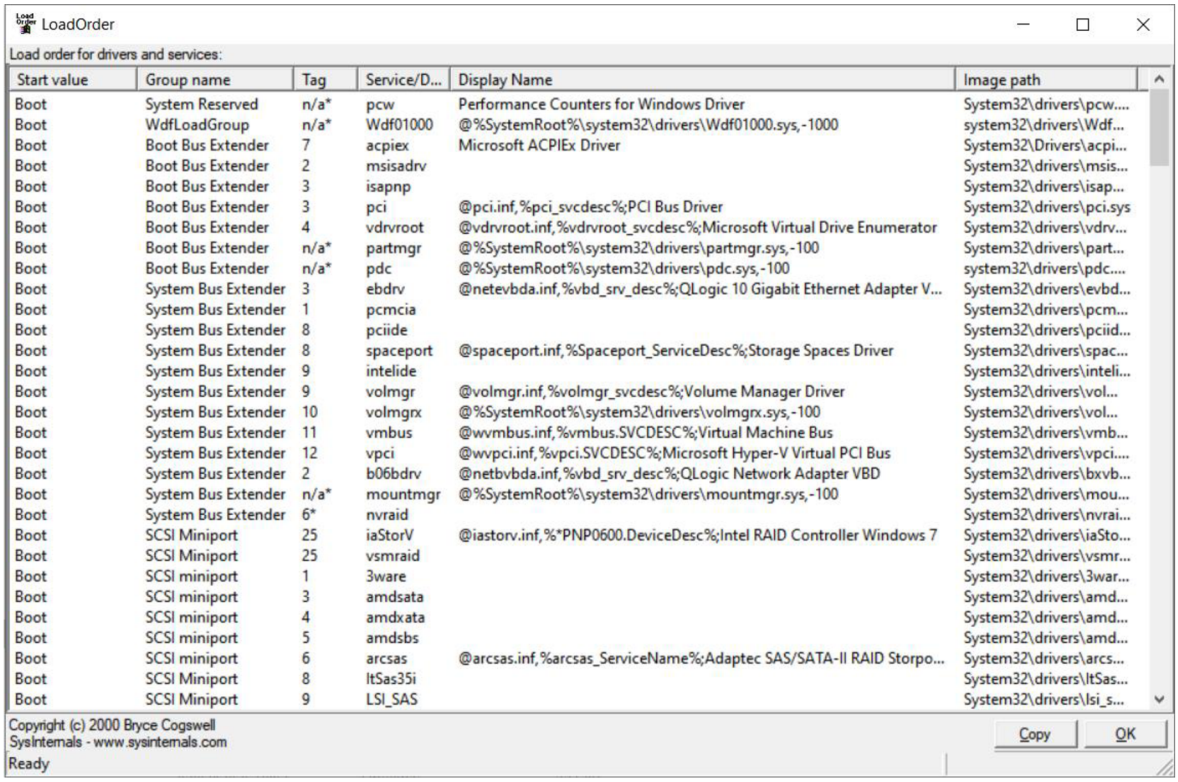
- **-u** – specifikace uživatelského jména vzdáleného počítače
- **-p** – zadání hesla daného uživatele
- **-s** – vypíše software instalovaný na daném zařízení
- **-d** – výpis informací o disku daného zařízení

- **@file** – výpis daných informací o všech zařízeních definovaných v textovém souboru, ke kterému musíme definovat cestu

- **-c** – export výpisu ve formátu CSV

3.3.3 LoadOrder

Jednoduchý nástroj zobrazující přibližné pořadí nahrávání jednotlivých ovladačů a služeb systému kteréhokoli systému Windows při jeho startu. Nevyžaduje administrátorská oprávnění.



Obrázek 55 – LoadOrder s grafickým výstupem

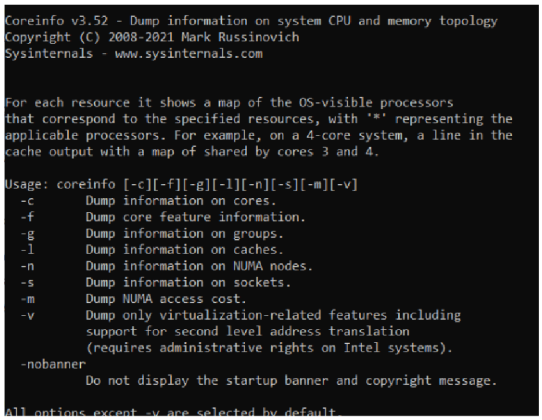
K dispozici je verze jak s grafickým výstupem (LoadOrd64), tak i s výstupem na konzolovou řádku (LoadOrdC64) stejně jako verze pro operační systémy založené na 32bitové architektuře.

3.3.4 CoreInfo

CoreInfo je aplikace pro konzolový řádek, která dokáže uživateli zobrazit mapování logických jader procesoru na jeho jádra fyzická, dále socket procesoru (ve kterém je zapojen), NUMA (Non-Uniform Memory Access) hodnotu – oboje pro PC/servery s více procesory zapojenými do několika patic. K získání těchto informací využívá jedno z API systémů Windows, konkrétně SysInfoAPI, z kterého volá funkci GetLogicalProcessorInformation()[13].

Při použití příkazu *coreinfo/coreinfo64* (u 64bitových systémů) dojde k výpisu takřka všech informací, které tento nástroj umí poskytnout

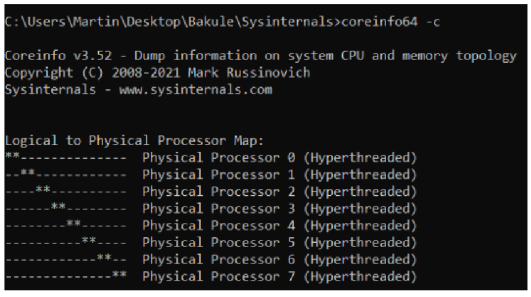
Možnosti nastavení (viz obr. 56) lze vypsat obligátním otazníkem jako argumentem, tedy „…*coreinfo64 -?*“.



Obrázek 56 – Coreinfo – nastavení

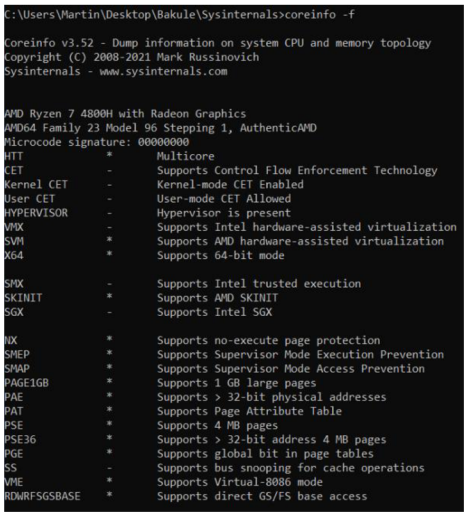
Pokud chceme pouze určitou část z tohoto výpisu, můžeme si příkaz nastavit. Jeho syntaxe pak bude např. „*C:\Sysinternals\coreinfo64 -X*“, kdy za X můžeme dosadit jeden z následujících argumentů [14]:

1. **-c**: omezí výstup programu na výpis namapování logických jader procesoru na jádra fyzická



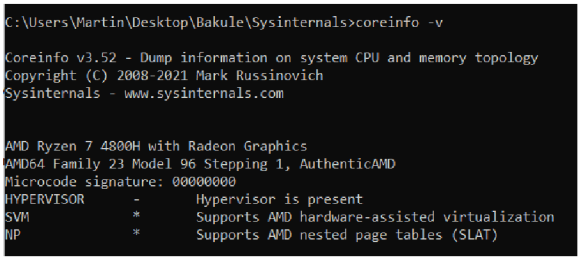
Obrázek 57 – Coreinfo -c

2. **-f**: argument – f nám vypíše, které z běžných funkcí náš procesor podporuje (typicky možnosti virtualizace, hyperthreading, instrukční sady a mnoho dalších [1-str.450-452]) – podporované jsou označeny hvězdičkou (*), nepodporované minusem (-)



Obrázek 58 – Coreinfo -f (výstup zkrácen)

3. **-v:** použití tohoto argumentu zobrazí pouze procesorem podporované virtualizační funkce

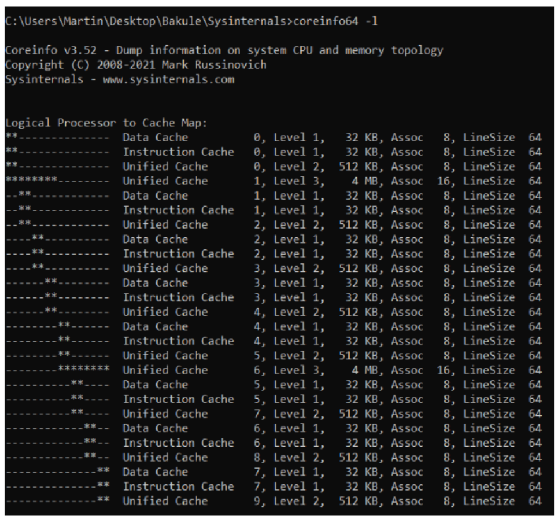


Obrázek 59 – Coreinfo -v

4. **-g**: výstupem je přiřazení (logických) jader procesoru ke skupině procesorů
    - na běžných osobních počítačích budou všechna v jedné – Group 0
    - více procesorových skupin připadá v úvahu až u počítačů s více než 64 jádry – typicky procesorová pole v serverech, pak začínají vznikat skupiny Group 1, Group 2 atd. [15]

5. **-l**: ve výpisu najdeme mapování jednotlivých logických jader procesoru k různým úrovním cache pamětí

*Na obrázku 60 pak můžeme vidět toto mapování pro mobilní procesor AMD Ryzen 4000 Series -> 2*32 KB (datová a instrukční) L1 cache na jádro, 512 KB L2 cache na jádro a 2* 4 MB L3 cache (každá pro 4 fyzická jádra)*



Obrázek 60 – Coreinfo64 -l

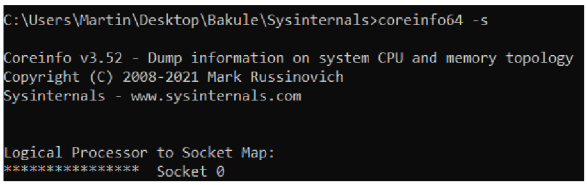
6. **-n**: zobrazí aktuální mapování jednotlivých jader procesoru k tzv. NUMA (Non-Uniform Memory Access) uzlům

    - NUMA je typ architektury sdružující jednotlivé procesory v multiprocesorovém systému do výše zmíněných uzlů za účelem optimalizace přístupu jednotlivých procesorů do sdílené paměti systému a s tím související zachování výkonu [16]

7. **-m**: tento výpis úzce souvisí s předchozí možností, informuje nás o rychlosti komunikace mezi jednotlivými procesorovými uzly na poměrné škále

    - Nejrychlejší spojení mezi uzly je označeno jako 1.0, ostatní jsou násobky této rychlosti

8. **-s**: spuštěním s tímto argumentem si zobrazíme mapování jednotlivých logických jader procesoru do jednotlivých fyzických socketů počítače



Obrázek 61 – Coreinfo64 -s

4 Praktické využití vybraných nástrojů

4.1 Vytvoření a nastavení virtuálního počítače

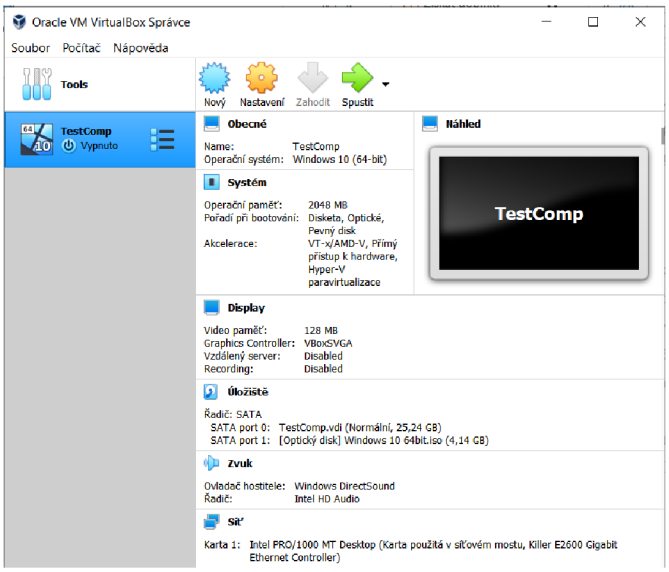
Než začneme s praktickými úlohami pro demonstraci použití nástrojů Sysinternals, vytvoříme si nejprve virtuální stroj. Pro mnoho funkcí jednotlivých programů je třeba mít administrátorská oprávnění, což u virtuálního stroje (na rozdíl od toho fyzického) není problém. Navíc jak bylo uvedeno u nástroje Autoruns (str. 37), neuváženým experimentováním s některými nástroji (těmi, které aktivně ovlivňují nastavení systému) si můžeme způsobit problémy s počítačem. A v neposlední řadě nám bude běžící virtuální systém simulovat vzdálený počítač, na kterém je třeba provést analýzu.

Nejprve musíme získat .iso obraz systému Windows. Předpokladem je, že máme k dispozici počítač s Windows 10 (případně verze 11). Mohli bychom použít jeden z dalších nástrojů Sysinternals, **Disk2vhd**. Ten dokáže vytvořit kopii Vašeho pevného disku a uložit ji pro použití v libovolném programu pro tvorbu virtuálních PC. Nicméně to vede k tomu, že pro jeho využití je třeba velké množství volného místa na disku. Proto se mu nyní vyhneme a použijeme **Media Creation Tool** od Microsoftu (návod pro jeho použití lze nalézt v knize Windows 10 Troubleshooting [17–str.43-45]). Nejprve si stáhneme tuto utilitu ze stránek Microsoftu (odkaz) a pak ji spustíme.

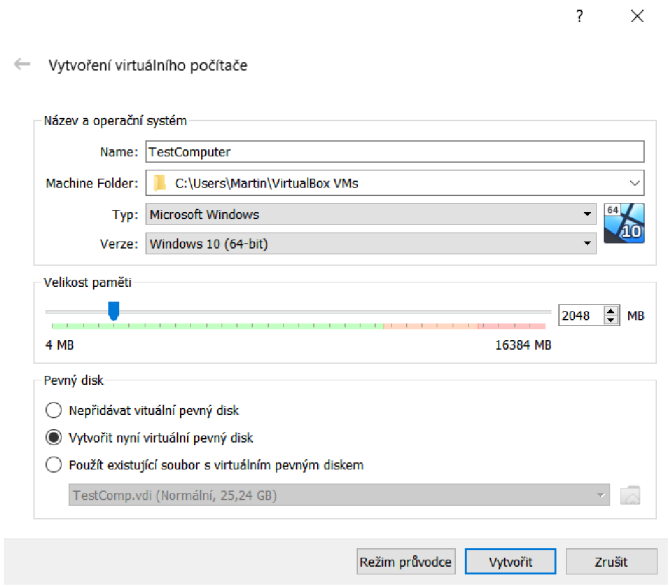
Po spuštění je třeba odsouhlasit licenční podmínky použití programu a následně zvolit možnost Vytvořit instalační média. V dalším kroku se zvolí jazyk, edice a architektura OS (implicitně jsou přednastaveny možnosti shodné s PC, na kterém spouštíme Media Creation Tool). Nakonec si vybereme možnost Soubor ISO a vybereme složku, kde se má tento obraz vytvořit. Po kliknutí na tlačítko Uložit začne stahování. Po jeho dokončení máme k dispozici .iso soubor, který můžeme dále používat.

Nyní můžeme přistoupit k vytvoření samotného virtuálního stroje. K tomu použijeme program VirtualBox od společnosti Oracle (ke stažení na stránkách virtualbox.org). Po jeho instalaci se nám otevře hlavní okno (obr. 62).

Nejprve klikněme na tlačítko **Nový**. Objeví se nám podokno s nastavením našeho virtuálního stroje (obr. 63). Vyplníme jméno nového počítače, dále určíme složku, ve které se má vytvořit virtuální stroj, určíme OS, který má být použit (v našem případě ponecháme implicitní Microsoft Windows) a nakonec jeho verzi a architekturu (např. Windows 10 64-bit). Ve spodní části klikneme na tlačítko Expertní režim. Okno se rozšíří o možnost volby velikosti paměti RAM, kterou bude mít počítač k dispozici (doporučeno alespoň 2 GB) a také o nastavení pevného disku virtuálního stroje. Zde ponecháme možnost Vytvořit nyní virtuální pevný disk. Kliknutím na tlačítko Vytvořit se otevře okno s upřesňujícími nastaveními virtuálního pevného disku nově vytvářeného VM. Zvolíme umístění složky disku VM, jeho maximální velikost, typ souboru (ponecháme .vdi) a typ alokace (dynamická vs pevná – ponecháme dynamickou – disk se bude zvětšovat do maximální velikosti postupně, jak bude třeba). Kliknutím na tlačítko Vytvořit vytvoříme virtuální stroj. Jak vidíme, byl přidán do seznamu v hlavním okně VirtualBoxu. Nyní ho můžeme spustit (dvojklikem nebo označením a stisknutím tlačítka Spustit). Při prvním spuštění je třeba projít standardní inicializací systému Windows – nastavení jazyka, klávesnice, uživatelského jména a hesla (případně přihlášení k účtu Microsoft). Po chvíli se dostáváme na plochu virtuálního počítače.



Obrázek 62 – VirtualBox – hlavní okno



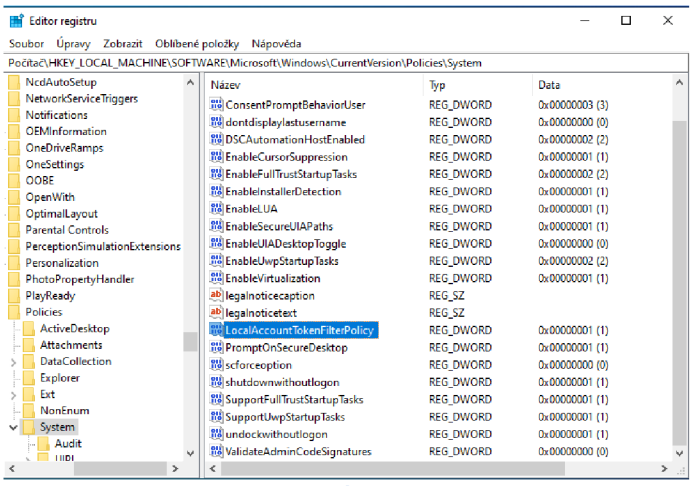
Obrázek 63 – Podokno nastavení VM

Nyní bychom mohli přejít k další úloze, nicméně pokud chceme, aby byl počítač přístupný analýze zvnějšku (a simuloval tak chování počítače např. ve firemní síti), je třeba provést příslušná nastavení (která významně snižují zabezpečení počítače a autor práce je nedoporučuje používat na jakémkoli jiném počítači).

Nejprve je třeba vypnout firewall počítače. V nabídce Start otevřeme okno Nastavení, klikneme na položku Aktualizace a zabezpečení, v nabídce vlevo klikneme na položku Zabezpečení Windows a následně vpravo na Firewall a ochrana sítě. Otevře se okno s nastavením firewallu. Zde je třeba vše vypnout.

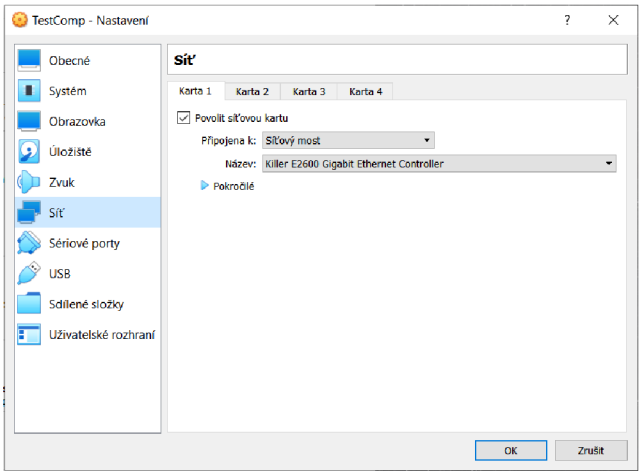
Dále nastavíme sdílení složek. V Exploreru otevřeme složku Tento počítač, klikneme pravým tlačítkem na disk C, najdeme možnost Udělit přístup pro a klikneme na možnost Rozšířené možnosti sdílení. V nově otevřeném podokně zaškrtneme možnost Sdílet tuto složku, níže klikneme na tlačítko Oprávnění, v dalším podokně označíme uživatele Everyone a ve sloupci povolit zaškrtneme možnost Úplné řízení.

Nakonec zrušíme nutnost potvrzování administrátorských oprávnění (UAC). Pomocí kláves **Win+R** otevřeme nabídku Spustit a zadáme příkaz *regedit*, kterým otevřeme Editor Registru. Zde v levém podokně najdeme složku na adrese **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System** (lze zadat tuto adresu do adresního řádku v horní části okna). Na složku klikneme pravým tlačítkem, najedeme na položku Nový a klikneme na možnost Hodnota DWORD (32bitová). V pravé části se objeví nový záznam. Jeho název změníme na **LocalAccountTokenFilterPolicy**, následně klikneme opět pravým tlačítkem a klikneme na položku **Změnit...**. Zde změníme hodnotu na 1. Výslednou podobu vidíme na obrázku 64.



Obrázek 64 – Úprava registru

Nakonec vypneme virtuální stroj, v hlavním okně VirtualBoxu klikneme na Nastavení, přejdeme na záložku Síť a připojení změníme z NAT na Síťový most.



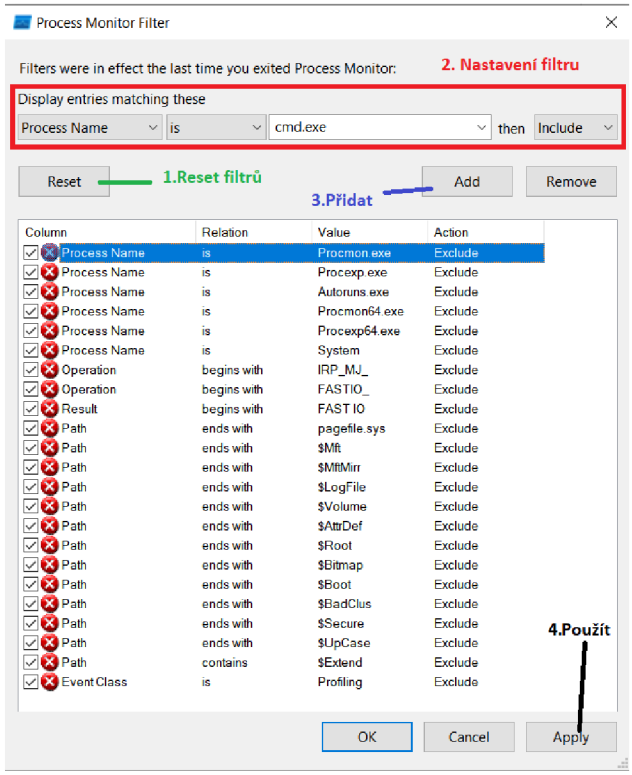
Obrázek 65 – Nastavení sítě VM

Těmito třemi kroky jsme eliminovali většinu problémů s připojením, které by mohli nastat při pokusu o připojení. Po dokončení této úlohy tedy máme zvnějšku přístupný virtuální počítač, na kterém si můžeme ukázat vzdálenou správu systému pomocí nástrojů vybraných nástrojů Sysinternals.

4.2 Jak interaguje vybraný proces se systémem? (Process Monitor)

V kapitole 3.1.2 jsme si představili jeden z nejpopulárnějších nástrojů Sysinternals, Process Monitor. Nyní si jeho použití předvedeme prakticky. Zjistíme, že počet interakcí, které musí běžně využívaný program provést, aby pro uživatele provedl zdánlivě jednoduchý úkon, může být překvapivý.

Ukažme si to na příkladu obyčejného „Hello World“ skriptu. Vytvoříme si obyčejný .txt soubor, napíšeme *echo Hello World* a uložíme jako soubor s příponou .cmd. Dále otevřeme Příkazový řádek, přejdeme do adresáře s vytvořeným skriptem a jeho jméno i s příponou napíšeme. Zatím nepotvrzujeme. Otevřeme okno Process Monitoru, který nám při startu zobrazí okno pro filtrování. Zde vidíme nastavení uložené z posledního spuštění programu. Kliknutím na tlačítko Reset filtry uvedeme do implicitního nastavení, kdy nejsou zachytávány události týkající se vybraných nástrojů Sysinternals, procesu System nebo události vztahující se k profilování procesů. V horní části nastavíme Process Name is cmd.exe *then* Include, tlačítkem Add tento filtr přidáme k ostatním, tlačítkem Apply připravené filtry aktivujeme kliknutím na OK přejdeme do hlavního okna.



Obrázek 66 – Filtrování událostí spojených s Příkazovým řádkem

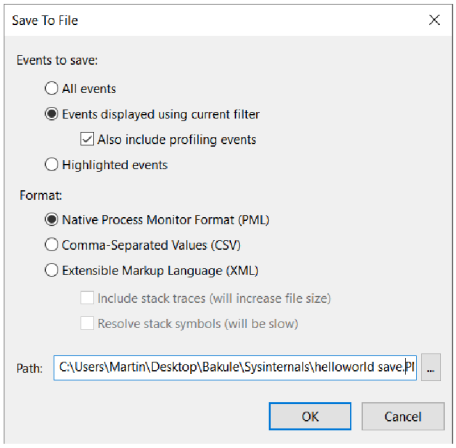
Nejprve se ujistíme, že v horní části je zaškrtnuto tlačítko Capture, tedy že dochází k záznamu (Příkazový řádek je nečinný a ostatní události filtrujeme – toto můžeme vidět v levém dolním rohu hlavního okna).

Dále se musíme rozhodnout, zda chceme zaznamenávat všechny události, nebo ty nevyhovující filtru rovnou zahazovat (k tomu bychom využili možnost Drop Filtered Events, viz str. 29).

Nyní přejdeme do okna Příkazové řádky a potvrdíme připravený příkaz.

I při provedení takto jednoduché úlohy dojde k záznamu zhruba stovky událostí, ze kterých můžeme vyčíst postup. V našem případě půjde postupně o vyhledání souboru ke spuštění, řešení synchronizace (zamknutí souboru) a čtení obsahu, nalezení konce souboru, jeho uvolnění a zavření.

Nyní si můžeme vytvořený záznam uložit např. pro pozdější analýzu. Klikneme na tlačítko Save (nebo si možnost Save najdeme v Menu) a objeví se nabídka uložení. Zde si opět můžeme zvolit, zda chceme ve vytvářeném souboru ponechat události, které byly zachyceny a neodpovídaly nastavenému filtru, dále typ souboru (nativní .pml, který lze opět otevřít v Process Monitoru nebo formáty .csv nebo .xml pro otevření v jiných programech (případně pro strojové zpracování) a nakonec cestu, kam bude soubor uložen.



Obrázek 67 – Nastavení uložení

Nicméně pojďme se podívat na další možné využití Process Monitoru. Představme si, že jako administrátorovi systému nám volá některý z uživatelů, že má problém s konkrétním programem. Abychom zjistili, co k danému problému vede, můžeme použít CLI možnosti Process Monitoru a ovládat ho pomocí příkazové řádky.

V kapitole věnované tomuto nástroji ale není zmínka o jeho použití na vzdáleném počítači (a nástroj to skutečně neumí), takže jak na to? Využijeme jeden z dalších nástrojů Sysinternals jménem **PsExec**, který nám umožňuje provádět příkazy na vzdáleném počítači. V našem případě využijeme v první úloze nakonfigurovaný virtuální stroj, který nám bude simulovat počítač vzdáleného uživatele.

Nejprve spustíme virtuální stroj a přihlásíme se. Následně na svém fyzickém počítači spustíme Příkazový řádek (s oprávněním administrátora) a přesuneme se do složky s rozbalenými nástroji balíku Sysinternals.
Zde zadáme následující příkaz:

- **psexec -sdc \\Testcomp procmon /accepteula /backingfile C:\Users\Marti\Documents\procmonlog.pml /quiet**

Nástroj psexec provede zadaný příkaz s následujícími argumenty [18]:

- -s = příkaz se na vzdáleném počítači provede s oprávněním admina
- -d = nečekat na ukončení procesu
- -c = zkopírovat spustitelný soubor do vzdáleného počítače a spustit ho
- \\Testcomp – název počítače v síti

Následuje určení aplikace ke zkopírování a spuštění, v našem případě Process Monitor. Využijeme některé z argumentů (viz str. 32):

- /accepteula – automatický souhlas s licenčním ujednáním Sysinternals (celé znění je v příloze)
- /backingfile <cesta k souboru\jméno souboru> - zaznamenané události se nebudou ukládat do paměti VM, ale rovnou do .pml souboru (viz str. 27), který se vytvoří v nastaveném adresáři a ponese zadané jméno
- /quiet – po spuštění Procmonu nebude třeba potvrzovat nastavení filtru

Nyní se vrátíme do prostředí virtuálního stroje a provedeme akci, kterou chceme monitorovat – pro ukázku stačí otevřít a zavřít prohlížeč Microsoft Edge. Zaznamenané události budou uloženy ve výše zmíněném souboru. Než přistoupíme k jeho zkopírování do našeho počítače k analýze, je třeba ukončit běžící Process Monitor:

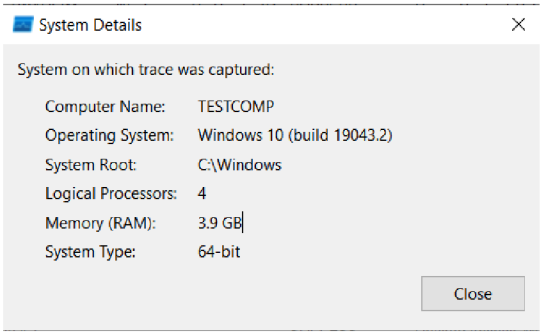
- **psexec -sd \\Testcomp procmon /accepteula /terminate/quiet**

Teď můžeme provést kopírování do námi vybrané složky (např. na plochu):

- **copy \\Testcomp\c$\Users\Marti\Documents\procmonlog.pml C:\Users\\Martin\Desktop**

Protože jde o soubor ve formátu .pml, můžeme si vytvořený soubor otevřít v Process Monitoru. Nastavíme si filtr (dle návodu výše), aby zobrazoval události procesu *msedge.exe* a potvrdíme.

Pokud bychom si chtěli ověřit, na jakém počítači byly tyto události zachyceny, najdeme v menu nabídku Tools a klikneme na první položku System Details (viz str. 29).



Obrázek 68 – Detaily o vzdáleném systému

4.3 Generování a analýza výpisu paměti vybraného procesu (ProcDump, WinDbg Preview)

Jak bylo uvedeno v teoretické části, nástroj ProcDump je používán například při ladění vyvíjených aplikací, kde jím vytvořené výpisy pomáhají s odhalováním bugů. Tyto chyby mohou ovlivňovat jak rychlost vyvíjené aplikace, tak rychlost (a ve finále i stabilitu) operačního systému Windows, na kterém jsou spuštěny. Tyto chyby mohou způsobovat výkyvy ve využití procesoru počítače nebo zapříčinit neuvolňování dříve používané paměti (a tím neustále rostoucí nároky na operační paměť). Toho můžeme využít při vytváření podmínek pro tvorbu výše zmíněného výpisu.

Abychom si toto mohli prakticky ukázat, potřebujeme něco jednoduchého, co nám bude generovat využití procesoru nebo rostoucí spotřebu paměti. A protože nemáme k dispozici vyvíjený software, co by nám danou situaci vytvářel, pomůžeme si uměle pomocí nástroje Sysinternals jménem **CPUStress64** (umělé vytížení CPU) a **Poznámkový blok** (rostoucí spotřeba paměti při vkládání množství textu).

Standardní syntaxe pro použití nástroje s implicitním nastavením je jednoduchá. V příkazové řádce zadáme příkaz *procdump* a název procesu, pro který chceme okamžitě vytvořit výpis paměti. Pokud je v systému spuštěna jen jedna instance daného procesu (program tedy najde jen jednu shodu), dojde k okamžitému vytvoření dump souboru. Bez dalšího upřesnění bude vytvořen jen minidump soubor (viz str. 34) ve stejném adresáři, odkud byl Procdump spuštěn. Název souboru je standardně ve formátu „*název procesu datum čas*“. Nicméně název souboru a místo, kam bude výpis uložen lze změnit prostým zadáním názvu souboru do příkazu.

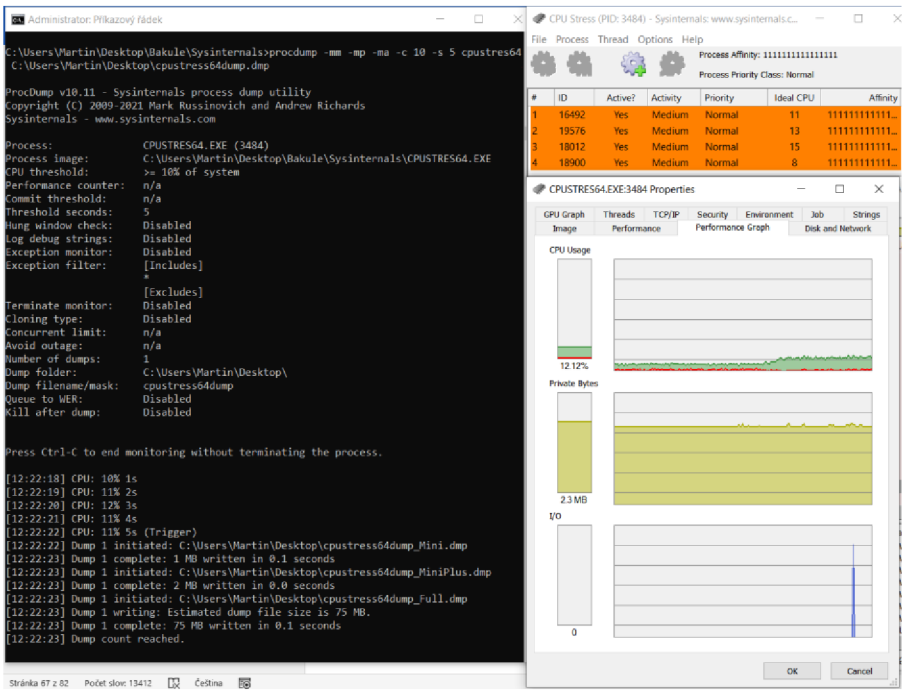
Nyní si vyzkoušejme tvorbu příkazů s argumenty. Začneme se záchytem při prahovém vytížení CPU. Otevřeme program CPUStress64 a kliknutím na tlačítko v horní části si vytvoříme několik vláken, se kterými budeme následně operovat. Vlákna nyní aktivujeme (implicitně po vytvoření jsou nečinná). Otevřeme **Process Explorer**, najdeme proces cpustres64.exe (můžeme použít vyhledávání tažením do okna – viz str. 19), dvojklikem na něj otevřeme okno s podrobnostmi o procesu a vybereme záložku Performance Graph. Zde pak můžeme sledovat aktuální vytížení procesoru sledovaným procesem. Při dostatečně rychlé obnovovací frekvenci dat (str. 18)

můžeme vidět chvilkovou změnu barvy pozadí daného procesu na šedou. Jde o důkaz toho, že proces byl po dobu záchytu pozastaven (ve stavu Suspended – str. 8, 35).

V příkazové řádce nyní můžeme zadat příkaz pro záchyt dané události:

- ***procdump -mm -mp -ma -c 10 -s 5 cpustres64 C:\Users\Martin\Desktop\ \cpustress64dump.dmp***

Pojďme si příkaz rozebrat. Nejprve pomocí argumentů určujeme typ výpisu paměti (rozdíly viz str.34), následně určíme horní práh vytížení CPU a pomocí argumentu -s počet sekund, po které musí daná situace trvat, než dojde k tvorbě výpisu. Následuje jméno procesu a nakonec složka, kam bude výpis uložen (včetně jeho jména). Příkaz potvrdíme, přejdeme do okna CPUStress a za stálého sledování výkonu na grafu v Process Exploreru zvyšujeme jejich aktivitu. Po dosažení prahu začne vidíme v okně příkazové řádky odpočet (při kterém se kontroluje, zda stále trvá situace pro záchyt) a nakonec potvrzení o vytvoření souborů výpisu paměti o různém obsahu (situaci lze vidět na obrázku 69). Od každého typu výpisu jsme získali jeden, jinak by se jejich počet měnil argumentem -n <počet>.



Obrázek 69 – Výřez obrazovky při záchytu

Další možností je monitoring procesů běžící na vzdáleném počítači, například na serveru. Postup bude podobný jako jsme si uvedli u Process Monitoru, tedy využijeme náš virtuální stroj a program PsExec. Dříve jsme si předvedli kopírování souboru ze vzdáleného počítače na fyzický, teď to jen otočíme a zkopírujeme soubor *procdump.exe* do libovolné složky virtuálního počítače.

Tentokrát budeme sledovat využití paměti procesem. Využijeme obyčejný poznámkový blok (notepad.exe), u kterého můžeme zvyšovat spotřebu paměti prostým vkládáním textu. V případě, že je na cílovém počítači spuštěno více procesů téhož jména, vyhodí Procdump chybu. Musíme tedy pomocí PID (Process ID) specifikovat, který má být monitorován.

Spustíme tedy VirtualBox a z něj náš VM. V něm si připravíme (otevřeme) Poznámkový blok. Na fyzickém stroji si otevřeme příkazový řádek, přesuneme se do složky s nástroji Sysinternals a otevřeme si příkazový řádek vzdáleného počítače příkazem:

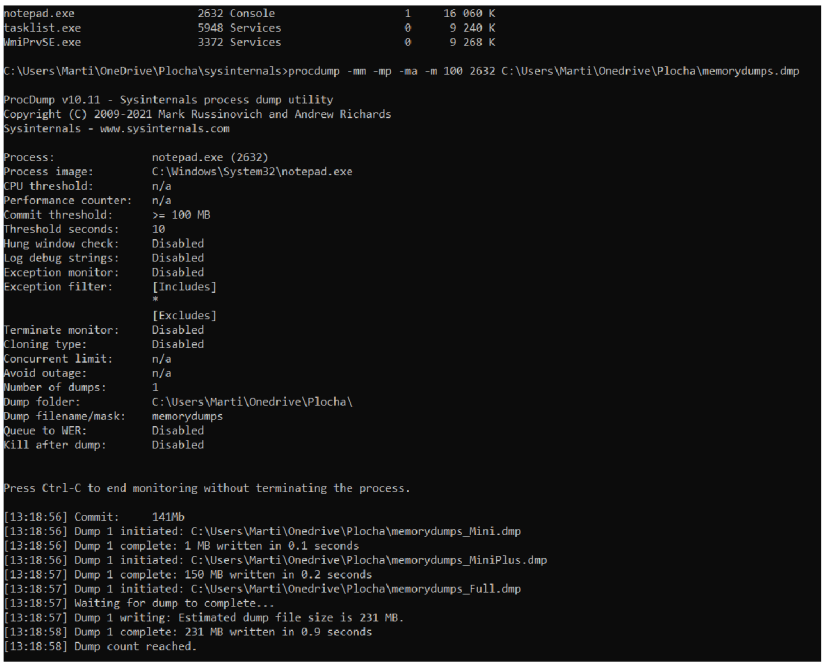
- ***psexec \\Testcomp cmd***

Nyní zadáme příkaz *tasklist*. Ten vypíše všechny na počítači běžící procesy i s jejich detaily, včetně PID.
Nyní můžeme spustit procdump. V příkazové řádce se přesuneme do složky se Sysinternals a zadáme následující příkaz:

- **procdump -mm -mp -ma -m 100 2632 C:\Users\temp\memorydump.dmp**

Syntaxe příkazu je tedy velmi podobná, liší se jen monitorovaná veličina (argument -m pro horní práh spotřeby paměti) a způsob určení cílového procesu. Pro notepad.exe v této PC session jsme zjistili PID 2632 (horní část obrázku 67). Pamatujme ale, že PID není konstantní, stejné PID může být přidělené jinému procesu a námi sledovaný proces může mít při příštím spuštění PC rovněž přidělené jiné (to je rozdíl oproti GUID, které je popsané na str. 46).

Na obrázku 70 vidíme, že přidáváním textu do Poznámkového bloku jsme dosáhli prahu využití paměti a došlo k záchytu všech 3 typů události. Tyto si nyní pomocí příkazu *copy* zkopírujeme do našeho fyzického PC a můžeme je analyzovat.

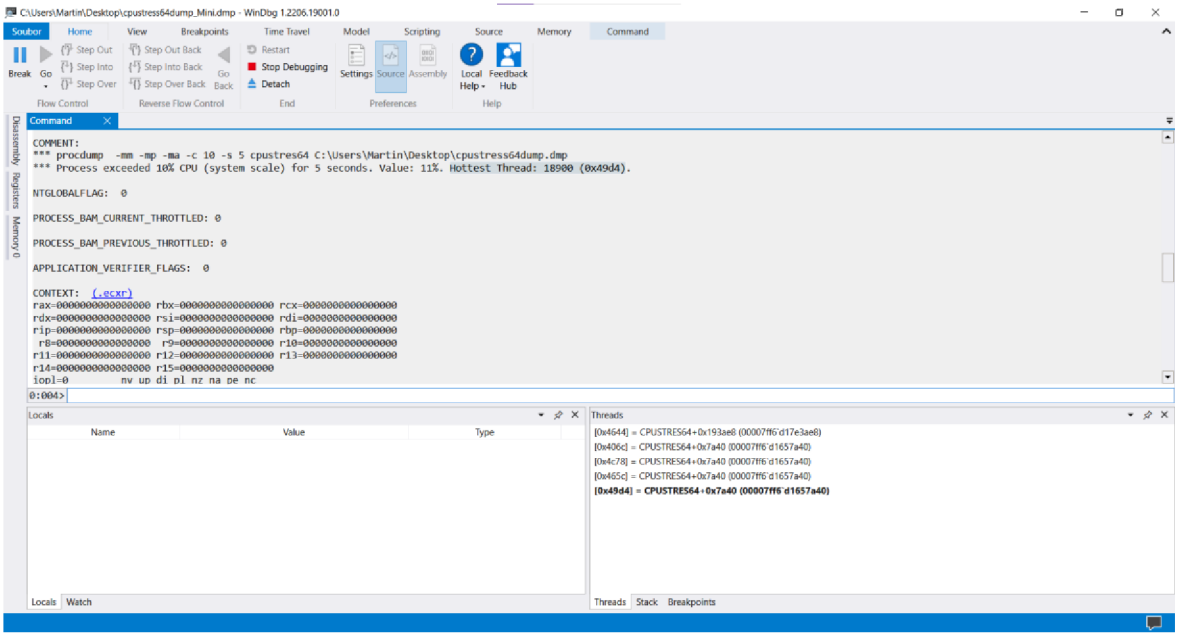


Obrázek 70 – Záchyt na vzdáleném počítači

Poté, co jsme získali výpisy paměti je můžeme otevřít. K tomu je třeba adekvátní program. Nejjednodušším způsobem je stáhnout a instalovat si do počítače program WinDbg Preview, který je zdarma dostupný na Microsoft Store (jde o novější variantu populárního WinDbg z balíčku Debugging Tools for Windows, navíc s instalací na jedno kliknutí). Po jeho instalaci je automaticky nastaven jako preferovaný program pro otevírání souborů s příponou .dmp. Více informací o jeho použití lze nalézt v dokumentaci [19].

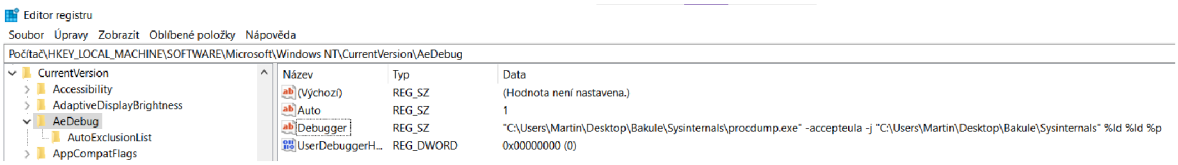
Nyní si otevřeme například výpis paměti vzniklý monitorováním programu CPUStress64. Pokud otevíráme WinDbg Preview poprvé, dojde k instalaci potřebných částí (znakové sady apod.). Následně klikneme na nabízený příkaz *!analyze -v*, čímž dojde k výpisu informací (přehled vláken, výpis ze zásobníku volání apod. – lze vidět na obrázku 71), které můžeme využít k procesu odstranění chyb nebo optimalizaci běhu programu.

Poznámka: Ideální je použít Procdump spolu s Process Monitorem (jak se lze dočíst na straně 22, události způsobené Procdumpem se zachycují jako Profiling Events). Získáme tak nejen stav při pádu procesu, ale i souhrn událostí, které k němu vedly, což nám pomůže při analýze a debuggingu.



Obrázek 71 – Výpis paměti analyzovaný ve WinDbg Preview

A konečně poslední využití nachází ProcDump jako automatický debugger v případě pádů programů. K tomu se využívá příkaz *procdump -i*, který provede instalaci a zaznamenání programu do registru systému – toto si lze ověřit v Editoru registru [1-str.201-202]. Otevřeme si složku (případně v adresním řádku zadáme):



Obrázek 72 – Editor registru s nastaveným Procdumpem

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\AeDebug, kde nalezneme hodnotu Debugger typu REG_SZ, která jako hodnotu má cestu ke spustitelnému souboru procdump.exe. Tímto způsobem vzniklé výpisy (jejichž typ lze nastavit pomocí argumentů stejně jako jsme to udělali u monitorovací funkce programu) pak nalezneme ve složce, kterou jsme buď specifikovali v příkazu nebo implicitně ve stejné složce, kde se nachází soubor procdump.exe. Názvy těchto výpisů nelze předem nastavovat a budou tak ve stejném formátu jako jsme si předvedli dříve.

V případě, že již nadále nechceme mít ProcDump nastavený jako automatický debugger, provedeme jeho odinstalaci příkazem *procdump -u*.

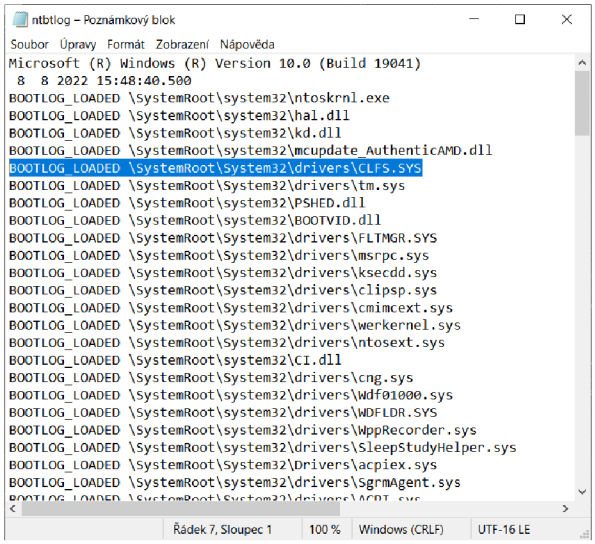
4.4 Analýza spouštění PC (Autoruns, Process Monitor)

V této úloze využijeme možností, které nabízí vybrané nástroje Sysinternals (konkrétně půjde o Process Monitor a Autoruns) k analýze spouštění počítače s operačním systémem Windows. Veškeré úkony budou probíhat na námi vytvořeném virtuálním stroji.

Standardním způsobem, jak zjistit, které ovladače jsou během startu systému nahrávány je nechat si vygenerovat Protokol spouštění (Boot Log). Kombinací kláves **Win+R** a příkazem *msconfig* otevřeme okno Konfigurace systému. Přejdeme na záložku Spuštění počítače a zaškrtneme možnost Protokol spouštění. Kliknutím na tlačítko Použít se objeví podokno s otázkou, zda chceme počítač restartovat ihned. Zatím vyčkáme.

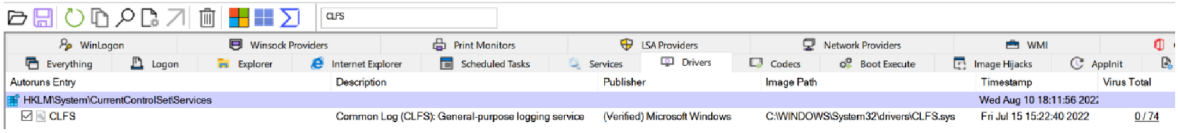
V protokolu spouštění najdeme jen pořadí a poznámku o úspěšnosti nahrávaných ovladačů. Chceme-li získat podrobnější výpis, je třeba využít další nástroj. Otevřeme program Process Monitor. V položce menu Options zvolíme možnost Enable Boot Logging, v nově otevřeném podoknu můžeme zvolit záchyt událostí typu Process Profiling (více na str. 22) a frekvenci jejich záznamu (0,1 s nebo 1 s). Potvrdíme a zavřeme Process Monitor. Následně můžeme náš virtuální stroj restartovat a posléze se opět přihlásit.

Vygenerovaný Protokol spouštění nyní můžeme nalézt ve složce **C:\Windows**, půjde o běžný textový soubor a ponese název *ntblog.txt*. Po jeho otevření vidíme výpis nahraných ovladačů včetně jejich umístění v souborovém systému. Výpis začíná nahráním ntoskrnl.exe (spouštěcí soubor jádra OS) a hal.dll (HW abstraktní vrstva). Zkusíme o jednotlivých ovladačích zjistit více. Otevřeme program Autoruns a přejdeme do záložky Drivers.



Obrázek 73 – Protokol spouštění

Zde můžeme vyhledávat detaily k jednotlivým ovladačům. Zkusíme hned první ovladač (na obrázku 73 označený CLFS.sys).



Obrázek 74 – Záznam o ovladač CLFS v Autoruns

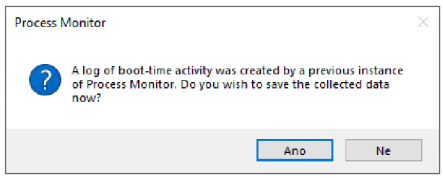
Povšimněme si jedné věci. Jak bylo uvedeno v kapitole věnované Autoruns, záznamy jsou vytvořeny podle obsahu Autostart Extension Point (lze nalézt na příslušné adrese v Editoru registru). Pokud v Protokolu spouštění najdeme záznam, jehož umístění není ve složce *Windows\System32\Drivers*, nebude se nám zobrazovat ani v okně Autoruns a nenajdeme ho ani v Editoru registru v příslušné složce Services (viz str.39). Tady můžeme pohodlně využít vyhledávání v záznamech a jednotlivé ovladače (soubory s příponou .sys a umístěné ve výše zmíněné složce) tu nalézt, včetně popisu jeho funkce, odkazu na umístění .sys souboru (jako v Protokolu spouštění), odkazu do Editoru registru apod. Získáme tak podrobnou představu o tom, co který z ovladačů, načítaných v průběhu startu OS vlastně dělá.

Pokud chceme dočasně pozastavit automatické spouštění některého z programů, služeb nebo ovladačů, stačí ho najít v seznamu (lze využít rychlé vyhledávání na liště tlačítek nebo některé z vyhledávacích podoken) a zrušit jeho označení. Tímto kliknutím dojde k vytvoření klíče AutorunsDisabled a přiřazení hodnoty true pro daný program (ovladač, službu aj.) v registrech systému Windows*. Můžeme si to ověřit, když označíme odškrtnutý řádek, klikneme pravým tlačítkem a pak zvolíme možnost Jump to Entry. Otevře se Editor registru (do kterého bychom se jinak dostávali přes kombinaci tlačítek Win+R a zkratkou regedit) s příslušným záznamem. Hodnotu uloženou v klíči AutorunsDisabled zde lze přepsat i manuálně, ale pohodlnější je opět využít okno programu Autoruns.

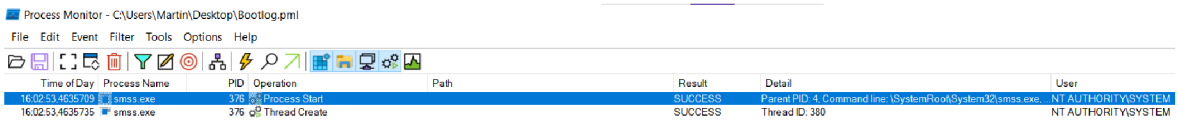
** Poznámka: Nedojde k jeho okamžitému vypnutí, ale bude zamezeno jeho startu při příští události, na kterou má reagovat (třeba přihlášení uživatele).*

Ačkoli v tomto případě jde o reverzibilní změnu, toto pozastavení může pořád snadno způsobit, že při příštím spuštění počítače nebude možné provést načtení operačního systému. Administrátor tak musí naprosto jistě vědět, co vlastně vypíná.

Nyní se podíváme na druhý záznam, který jsme si nechali vytvořit. Opět spustíme Process Monitor. Tentokrát je úvod jiný, nenabízí nám nastavení filtrů, ale uložení záznamu událostí, které proběhly během bootování počítače. Zvolíme formát (ponecháme .pml), název souboru a adresář, kam bude uložen. Samotné ukládání může trvat i několik minut, je doprovázeno grafickou lištou znázorňující aktuální postup. Také je nutné se ubezpečit, že máme k dispozici dostatečné množství místa na disku, protože soubor bude obsahovat cca 3 miliony zaznamenaných událostí a mít velikost kolem 1,3 GB. Po dokončení se v hlavním okně objeví výpis zachycených událostí.

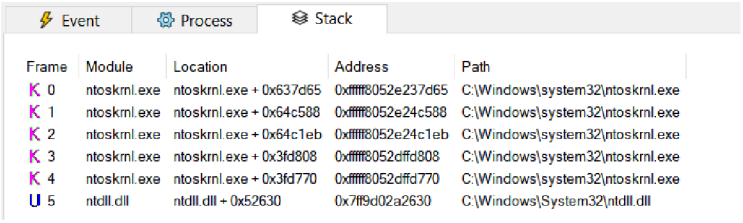


Obrázek 75 – Výzva po opětovném spuštění Process Monitoru



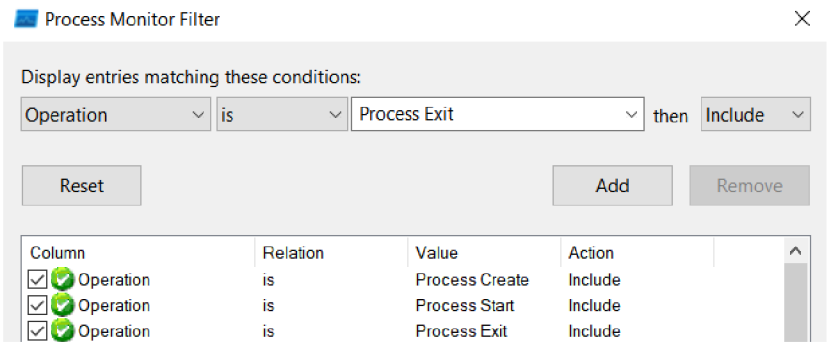
Obrázek 76 - Záznam startu – první proces

Můžeme vidět, že celý záznam začíná vytvořením procesu smss.exe (Session Manager [2-str.92-95] – Správce relací). Tento proces je synovským procesem procesu System (PID 4) a vůbec prvním, který při startu systému vykonává kód v uživatelském režimu (můžeme vidět ve výpisu ze zásobníku volání na obr.77).



Obrázek 77 – Výpis zásobníku volání smss.exe

Vzhledem k velkému množství událostí by bylo jejich efektivní procházení zdlouhavé. Přistoupíme tedy k vytvoření filtrů. Řekněme, že nás zajímá jen posloupnost vytváření a ukončování procesů při startu operačního systému. Otevřeme si tedy okno pro vytváření filtrů a zadáme podmínky podle obrázku 77.



Obrázek 78 – Filtry pro posloupnost procesů

Poznámka: Protože jsme nechali vytvářet Protokol spouštění vedle záchytu událostí Process Monitorem, můžeme zde najít také operace s jeho vytvářením spojené.

Time of Day	Process Name	PID	Operation	Path
16:02:53,4635709	smss.exe	376	Process Start	
16:02:56,5480291	smss.exe	376	Process Create	C:\Windows\system32\autochk.exe
16:02:56,5480332	autochk.exe	400	Process Start	
16:02:56,5924856	autochk.exe	400	Process Exit	
16:02:56,9591246	smss.exe	376	Process Create	C:\Windows\System32\smss.exe
16:02:56,9591287	smss.exe	460	Process Start	
16:02:57,0761241	smss.exe	460	Process Create	C:\Windows\system32\csrss.exe
16:02:57,0761287	csrss.exe	468	Process Start	
16:02:57,2069246	smss.exe	376	Process Create	C:\Windows\System32\smss.exe
16:02:57,2069297	smss.exe	536	Process Start	
16:02:57,2077305	smss.exe	460	Process Create	C:\Windows\system32\wininit.exe
16:02:57,2077342	wininit.exe	544	Process Start	
16:02:57,2121398	smss.exe	460	Process Exit	
16:02:57,2168656	smss.exe	536	Process Create	C:\Windows\system32\csrss.exe
16:02:57,2168696	csrss.exe	560	Process Start	
16:02:57,2593275	smss.exe	536	Process Create	C:\Windows\system32\winlogon.exe
16:02:57,2593318	winlogon.exe	644	Process Start	
16:02:57,2675577	smss.exe	536	Process Exit	
16:02:57,2884060	wininit.exe	544	Process Create	C:\Windows\system32\services.exe
16:02:57,2884103	services.exe	680	Process Start	
16:02:57,3392494	wininit.exe	544	Process Create	C:\Windows\system32\lsass.exe
16:02:57,3392678	lsass.exe	712	Process Start	
16:02:57,5390012	services.exe	680	Process Create	C:\Windows\system32\svchost.exe
16:02:57,5390061	svchost.exe	824	Process Start	
16:02:57,5567598	winlogon.exe	644	Process Create	C:\Windows\system32\fontdrvhost.exe
16:02:57,5567669	fontdrvhost.exe	848	Process Start	
16:02:57,5568213	wininit.exe	544	Process Create	C:\Windows\system32\fontdrvhost.exe
16:02:57,5568253	fontdrvhost.exe	856	Process Start	
16:02:57,7177205	services.exe	680	Process Create	C:\Windows\system32\svchost.exe
16:02:57,7177252	svchost.exe	948	Process Start	
16:02:57,7529318	services.exe	680	Process Create	C:\Windows\system32\svchost.exe
16:02:57,7529589	svchost.exe	988	Process Start	
16:02:57,8519117	winlogon.exe	644	Process Create	C:\Windows\system32\dwm.exe
16:02:57,8519158	dwm.exe	412	Process Start	
16:02:57,8525277	winlogon.exe	644	Process Create	C:\Windows\system32\LogonUI.exe
16:02:57,8525321	LogonUI.exe	424	Process Start	
16:02:58,0065972	services.exe	680	Process Create	C:\Windows\system32\svchost.exe
16:02:58,0066035	svchost.exe	400	Process Start	
16:02:58,0211158	services.exe	680	Process Create	C:\Windows\system32\svchost.exe
16:02:58,0211215	svchost.exe	1040	Process Start	
16:02:58,0219867	services.exe	680	Process Create	C:\Windows\System32\svchost.exe
16:02:58,0219922	svchost.exe	1060	Process Start	
16:02:58,0572942	services.exe	680	Process Create	C:\Windows\System32\svchost.exe
16:02:58,0573001	svchost.exe	1116	Process Start	
16:02:58,0577112	services.exe	680	Process Create	C:\Windows\system32\svchost.exe
16:02:58,0577189	svchost.exe	1124	Process Start	
16:02:58,1025012	services.exe	680	Process Create	C:\Windows\system32\svchost.exe
16:02:58,1026641	svchost.exe	1232	Process Start	

Obrázek 79 – Výsledná posloupnost procesů

A podle tohoto výpisu se můžeme prakticky přesvědčit o posloupnosti vytváření procesů v operačním systému Windows 10 (postupně vytvářeny Session Managerem):

- Nejprve je vytvořen a spuštěn autochk.exe (nástroj pro kontrolu disků) [20]
- Následuje proces csrss.exe (Client/Sever Runtime Subsystem)
- Poté wininit.exe následovaný winlogon.exe

Wininit poté vytvoří proces services.exe (který následně vytváří vůbec první proces svchost.exe) a hned po něm lsass.exe, což je autentizační proces pro přihlášení uživatele do systému a tak dále.

4.5 Dlouhodobý monitoring počítače (System Monitor)

Jak jsme se mohli dočíst v kapitole věnované nástroji Sysmon (str. 45), po své instalaci a konfiguraci dokáže zaznamenávat různé události, které v počítači nastanou. Pojďme si nyní prakticky ukázat, jak toho dosáhnout. Opět můžete pracovat na fyzickém počítači nebo využít virtuální stroj vytvořený v úloze 1.

Nejprve provedeme instalaci do počítače:

- Otevřeme si *Příkazový řádek.*
- Dále provedeme změnu aktivního adresáře do složky s nástroji Sysinternals, například *cd .\Desktop\Sysinternals.*
- A zadáme příkaz „*Sysmon64 – i*“, kterým provedeme instalaci

Po této jednoduché instalaci ale bude Sysmon zaznamenávat jen vytváření a ukončování procesů. Pokud chceme zaznamenávat události jiného typu, musíme takto nainstalovanému programu dodat konfigurační soubor typu **.xml**. Jeho ukázku můžeme vidět na obrázku 65 nebo po zadání příkazu *Sysmon64 -? config*, kde lze najít vícero podobných ukázek.

```
A simple configuration xml file looks like this:

<Sysmon schemaversion="4.70">
  <!-- Capture all hashes -->
  <HashAlgorithms>*</HashAlgorithms>
  <EventFiltering>
    <!-- Log all drivers except if the signature -->
    <!-- contains Microsoft or Windows -->
    <DriverLoad onmatch="exclude">
      <Signature condition="contains">microsoft</Signature>
      <Signature condition="contains">windows</Signature>
    </DriverLoad>
    <!-- Do not log process termination -->
    <ProcessTerminate onmatch="include" />
    <!-- Log network connection if the destination port equal 443 -->
    <!-- or 80, and process isn't InternetExplorer -->
    <NetworkConnect onmatch="include">
      <DestinationPort>443</DestinationPort>
      <DestinationPort>80</DestinationPort>
    </NetworkConnect>
    <NetworkConnect onmatch="exclude">
      <Image condition="end with">iexplore.exe</Image>
    </NetworkConnect>
  </EventFiltering>
</Sysmon>
```

Obrázek 80 – Příklad konfiguračního souboru v programu

Pojďme si nyní jeden takový soubor vytvořit. Můžeme použít obyčejný textový editor, ale lépe bude použít IDE, například Visual Studio Code. Všechna pravidla v souboru jsou uvnitř značek:

<Sysmon schemaversion=“verze_schématu_pro_vaší_verzi_Sysmonu“> </Sysmon>

Následuje uvedení hashovacích algoritmů, stačí uvést mezi značky <HashAlgorithms> </HashAlgorithms> regulární výraz * pro libovolné.

Nyní vytvoříme pomocí značek *<EventFiltering> </EventFiltering>* místo pro vkládání pravidel a můžeme je začít vytvářet.

Pokud chceme, můžeme pravidla sdružovat do skupin:

- Pomocí značek *<RuleGroup name="jméno_skupiny_pravidel" groupRelation="or"> </RuleGroup>* vytvoříme skupinu. Argument groupRelation nám řeší vztahy mezi jednotlivými pravidly skupiny (zde logický OR – implicitní volba od Sysmon verze 9).

Nyní můžeme do skupiny vložit typ události daného pravidla. Jako značka se používá název pro některou ze skupin zaznamenaných událostí, jejichž výčet najdeme v dokumentaci [10].
Jak bylo uvedeno výše, Sysmon implicitně zaznamenává všechny vytvořené a ukončené procesy. Pokud chceme mít přehled, tak nejprve vložíme příkazy, které tomuto zabrání:

- *<ProcessCreate onmatch="exclude"/>*
- *<ProcessTerminate onmatch="exclude"/>*

Následně vložíme typ situace, kdy má dojít k tvorbě záznamu. Ten se liší podle typu události, půjde například o start spustitelného souboru *(<Image>*), konkrétní příkaz z příkazové řádky (*<CommandLine>*), cílová IP (*<DestinationIp>*), využívaný port (*<DestinationPort>*) a podobně. Takovému záznamu pak můžeme napsat jméno (*name=""*), které se bude zobrazovat v Prohlížeči událostí (Event Viewer) u záznamů vzniklých podle daného pravidla a logický argument *condition=""*, do kterého vložíme některou z následujících hodnot (kompletní výčet opět v dokumentaci [10], zde jen vybrané):

- *Is*: rovnost hodnot (název procesu, číslo portu, …)
- *Image*: ověřuje cestu ke spustitelnému .exe souboru nebo jeho jméno
- *Contains*: obsahuje daný řetězec
- *Begin with*: začíná touto hodnotou
- *Excludes*: neobsahuje tuto hodnotu
- *Less/more than*: lexikografické porovnání řetězce

Pojďme si některá základní pravidla ukázat:

Začneme u obyčejného **vytvoření procesu**, tedy totéž, co zaznamenává Sysmon už od instalace. My si ale můžeme stanovit, kterých procesů se má záznam týkat – události procesů splňujících podmínky mohou být buď zahrnuty (include), nebo naopak vynechány (exclude).

Nejprve uvedeme pravidlo odpovídajícími značkami a jak s ním bude Sysmon nakládat, v tomto případě *<ProcessCreate onmatch="include"> </ProcessCreate>*. Mezi ně pak můžeme vkládat značky uvozující jednotlivé podmínky. Vždy budou obsahovat informaci o tom, k čemu se váží (příkaz z příkazové řádky – *CommandLine*, spustitelný soubor – *Image*, …) a logický argument condition. Pro lepší orientaci lze přidat jméno pravidla. Například podle jména procesu půjde o:

<Image name="pravidlo" condition="image"> jméno_spustitelného_souboru </Image>.

Další možností je **záchyt síťových spojení**. Řekněme, že chceme zachytit každé SSH nebo Telnet spojení z nebo do počítače a třeba spojení se stránkami z české domény .cz. Daná síťová pravidla můžeme opět umístit do adekvátní skupiny (*<RuleGroup>*) a uvést značkou *<NetworkConnect onmatch="include"> </NetworkConnect>.*

Pokud chceme zachytit komunikaci probíhající přes protokoly Telnet a SSH (a implicitně využívající porty 22 a 23), použijeme pro pravidlo značky *<DestinationPort> </DestinationPort>*. Dále postupujeme stejně jako u procesu – můžeme určit jméno pravidla, povinně dát podmínku a mezi značky umístit danou hodnotu.

Pro české stránky pak pravidlo bude mezi značkami *<DestinationHostname></DestinationHostname>*. Jako podmínku určíme „contains" a do místa pro hodnotu napíšeme .cz.

config.xml

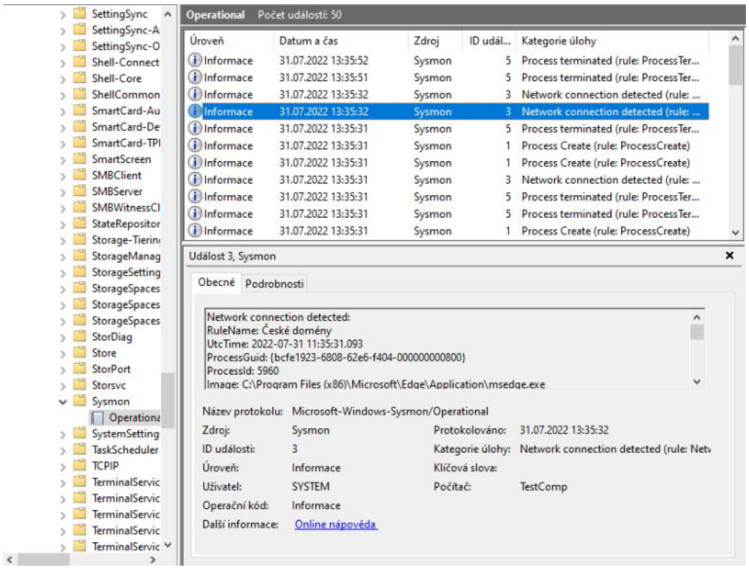
C: > Users > Martin > Desktop > config > config.xml

```
<Sysmon schemaversion="4.70">
    <HashAlgorithms>*</HashAlgorithms>
    <EventFiltering>
    <!-- Nyní můžeme přidávat typy jednotlivých událostí, jak jsou uvedeny v dokumentaci -->
        <RuleGroup name="Prohlížeče" groupRelation="or">
        <!-- Nejprve vynulujeme implicitní filtr Sysmonu -->
            <ProcessCreate onmatch="exclude"/>
            <ProcessTerminate onmatch="exclude"/>
            <!-- A vkládáme pravidla -->
            <ProcessCreate onmatch="include">
                <Image name="Firefox" condition="image">firefox.exe</Image>
                <Image name="Edge" condition="end with">msedge.exe</Image>
                <Image name="Internet Explorer" condition="contains">iexplore.exe</Image>
            </ProcessCreate>
        </RuleGroup>
        <RuleGroup name="Síťová spojení" groupRelation="or">
            <NetworkConnect onmatch="include">
                <DestinationPort name="SSH Spojení" condition="is">22</DestinationPort>
                <DestinationPort name="Telnet Spojení" condition="is">23</DestinationPort>
                <DestinationHostname name="České domény" condition="contains">.cz</DestinationHostname>
            </NetworkConnect>
        </RuleGroup>
    </EventFiltering>
</Sysmon>
```

Obrázek 81 – Vytvořený konfigurační soubor ve Visual Studio Code

Takto vytvořený konfigurační soubor nyní nahrajeme do Sysmonu. Použijeme příkaz **sysmon64 -c cesta_k_souboru**. Aktuálně užívanou konfiguraci si můžeme zobrazit příkazem **sysmon64 -s**.

Pokud Sysmon běží a využívá námi vytvořenou konfiguraci, můžeme se podívat na zachycené události. Otevřeme si Prohlížeč událostí (Event Viewer) systému Windows a v levém panelu najdeme jeho složku (viz str. 46). Nyní zde nalezneme pouze záznamy o událostech odpovídající pravidlům v konfiguračním souboru (viz obr. 67).



Obrázek 82 – Detail zachycené události

Pokud chceme konfiguraci vymazat a přejít zpět k implicitnímu nastavení, využijeme příkaz **sysmon64 -c**. Nakonec, pokud bychom chtěli Sysmon z počítače odinstalovat, použijeme příkaz **sysmon64 -u**.

5 Shrnutí výsledků

V této práci byly představeny nástroje, které dle mínění autora této práce představují to nejzajímavější z procesních, bezpečnostních a informačních nástrojů balíku Sysinternals Suite. Během výběru těchto nástrojů bylo zjištěno, že některé jsou již morálně zastaralé Příkladem může být třeba Port Monitor [21] (PortMon), nástroj pro sledování stavu sériových a paralelních portů PC (které byly z osobních počítačů vytlačeny rozhraním USB) nebo na straně 37 popisovaný RootkitRevealer. Během literární rešerše navíc autor práce zjistil nedostatek českojazyčných zdrojů, vycházel proto převážně ze zdrojů zahraničních.

Dále je třeba připomenout, že také vybrané nástroje představené v této práci podléhají kontinuálnímu vývoji a pravidelným aktualizacím, které jsou vždy oznamovány na oficiálním twitterovém účtu [22] a přehledné články, co je v daných nástrojích po aktualizaci nového, vychází na oficiálním blogu [23]. Proto se některá zjištění autora této práce liší od poznatků uvedených v některých starších zdrojích (rozdíly popsané například na stránkách 21 nebo 25), a proto je také možné, že pokud budou v budoucnu některé nástroje opět upraveny (případně z balíku přímo odstraněny) nebo budou nové vytvořeny, přestanou být poznatky uvedené v této práci aktuální, respektive nebudou již zcela přesné.

6 Závěry a doporučení

V teoretické části jsme si představily ty z pohledu autora této práce nejzajímavější a nejkomplexnější nástroje z balíčku Sysinternals Suite – dle zadání této bakalářské práce ze skupin Process Utilities, Security Utilities a System Information Utilities. V praktické části jsme si následně vytvořili virtuální počítač s operačním systémem Windows 10 a předvedli si jejich použití v situacích, které napodobují běžnou praxi.

Zkoumání by šlo nyní zaměřit na v této bakalářské práci neprozkoumané skupiny nástrojů Sysinternals – Network Utilities (např. TCPView), správu Active Directory nebo skupinu File & Disc Utilities (např. DiskMon, Contig, AccessChk a další).

Dalším zajímavým postupem ve zkoumání možností nástrojů Sysinternals by bylo vytvoření dobře izolovaného virtuálního počítače s operačním systémem Windows (ať už verze 10 nebo 11), jeho umělé infikování různými typy malwaru, ať už by šlo o různé viry – keyloggery, trojské koně, ransomware, síťové červy apod. Tento škodlivý kód je v době psaní této bakalářské práce k dispozici na platformě GitHUB v rámci projektu zvaného theZOO (odkaz zde), výhradně jen k edukativním účelům (autoři projektu nenesou odpovědnost při zneužití těchto kódů k jakékoli formě kyberkriminality)!

Na takto infikovaném počítači by pak bylo možné pozorovat konkrétní kroky toho kterého viru v systému, ať už pomocí nástroje Autoruns (zavedení virů ještě před dokončením startu operačního systému, maskování za jiné procesy), Process Explorer, Sysmon, Process Monitor (monitoring interakcí škodlivého softwaru se systémem) a dalších.

A konečně za třetí by bylo možné zaměřit se na automatizovanou správu operačních systémů, tedy zkoumat možnosti CLI verzí nástrojů SysInternals při tvorbě pokročilých skriptů v prostředí Powershell.

7 Seznam použité literatury

1. *RUSSINOVICH, Mark, MARGOSIS Aaron. Troubleshooting with the Windows Sysinternals Tools. 2nd edition. Redmond, WA: Microsoft Press, 2016. ISBN 978-0-7356-8444-7*

2. *YOSIFOVICH, Pavel, IONESCU Alex, RUSSINOVICH Mark, SOLOMON David A. Windows Internals: Part 1. 7th Edition. Redmond, WA: Microsoft Press, 2017. ISBN 978-0-7356-8418-8.*

3. *ALLIEVI, Andrea, Alex IONESCU, Mark E. RUSSINOVICH a David A. SOLOMON. Windows Internals: Part 2. Seventh edition. London, UK: Pearson Education, 2021. ISBN 978-0-13-546240-9.*

4. *SELECKÝ, Matúš. Windows Sysinternals: Vyladťe si systém. Brno: Albatros Media, 2013. ISBN 978-80-251-3823-6.*

5. *RUSSINOVICH, Mark, MIHAIUC Alexandru et al. ProcDump v10.11. Microsoft Docs [online]. Redmond, WA: Microsoft Press, 2021, 08/18/2021 [cit. 2022-02-02]. Dostupné z: https://docs.microsoft.com/en-us/sysinternals/downloads/procdump*

6. *RICHARDS, Andrew. MINIDUMP_CALLBACK_ROUTINE callback function (minidumpapiset.h). Microsoft Docs [online]. Redmond, WA: Microsoft Press, 2021, 13. 10. 2021 [cit. 2022-06-29]. Dostupné z: https://docs.microsoft.com/cs-cz/windows/win32/api/minidumpapiset/nc-minidumpapiset-minidump_callback_routine*

7. RUSSINOVICH, Mark. Sony, Rootkits and Digital Rights Management Gone Too Far. *Windows Blog Archive* [online]. Redmond, WA: Microsoft Press, 2005, Oct 31, 2005 [cit. 2022-06-20]. Dostupné z: https://techcommunity.microsoft.com/t5/windows-blog-archive/sony-rootkits-and-digital-rights-management-gone-too-far/ba-p/723442

8. *UROZ, Daniel a Ricardo J. RODRÍGUEZ. Characteristics and detectability of Windows auto-start extensibility points in memory forensics. Digital Investigation [online]. 2019, 28, S95-S104 [cit. 2022-07-13]. ISSN 17422876. Dostupné z: doi: 10.1016/j.diin.2019.01.026*

9. *RUSSINOVICH, Mark. Autoruns for Windows v14.09. Microsoft Docs [online]. Redmond, WA: Microsoft Press, 2022, February 16, 2022 [cit. 2022-07-12]. Dostupné z: https://docs.microsoft.com/en-us/sysinternals/downloads/autoruns*

10. *GARNIER, Thomas a Mark RUSSINOVICH. Sysmon v13.34. Microsoft Docs [online]. Redmond, WA: Microsoft Press, 2022, May 11, 2022 [cit. 2022-07-09]. Dostupné z: https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon*

11. *RUSSINOVICH, Mark. RAMMap v1.61. Microsoft Docs [online]. Redmond, WA: Microsoft, 2022, May 11, 2022 [cit. 2022-07-01]. Dostupné z: https://docs.microsoft.com/en-us/sysinternals/downloads/rammap*

12. *RUSSINOVICH, Mark a Helder MAGALHÃES. PsInfo v1.78. Microsoft Docs [online]. Redmond, WA: Microsoft Press, 2016, June 29, 2016 [cit. 2022-06-19]. Dostupné z: https://docs.microsoft.com/en-us/sysinternals/downloads/psinfo*

13. *GetLogicalProcessorInformation function (sysinfoapi.h). Microsoft Docs [online]. Redmond, WA: Microsoft Press, 2021, 13. 10. 2021 [cit. 2022-02-13]. Dostupné z: https://docs.microsoft.com/cs-cz/windows/win32/api/sysinfoapi/nf-sysinfoapi-getlogicalprocessorinformation?redirectedfrom=MSDN*

14. *RUSSINOVICH, Mark, YOSIFOVICH, Pavel. Coreinfo v3.52. Microsoft Docs [online]. Redmond, WA: Microsoft Press, 2021, February 22, 2021 [cit. 2022-02-13]. Dostupné z: https://docs.microsoft.com/en-us/sysinternals/downloads/coreinfo*

15. *BRIDGE, Karl, Kent SHARKEY, COULTER David et al. Processor Groups. Microsoft Docs [online]. Redmond, WA: Microsoft Press, 2021, 12/30/2021 [cit. 2022-02-17]. Dostupné z: https://docs.microsoft.com/en-us/windows/win32/procthread/processor-groups*

16. *BRIDGE, Karl, Zoe LIU a Quinn RADICH. NUMA Support. Microsoft Docs [online]. Redmond, WA: Microsoft Press, 2021, 08/19/2021 [cit. 2022-02-17]. Dostupné z: https://docs.microsoft.com/en-us/windows/win32/procthread/numa-support*

17. *HALSEY, Mike. Windows 10 Troubleshooting. New York, NY: Apress Media, 2016. ISBN 978-1-4842-0926-4.*

18. *RUSSINOVICH, Mark. PsExec v2.40. Microsoft Docs [online]. Redmond, WA: Microsoft Press, 2022, July 19, 2022 [cit. 2022-08-07]. Dostupné z: https://docs.microsoft.com/en-us/sysinternals/downloads/psexec*

19. *MARSHALL, Don a Amy VIVIANO. Debugging Using WinDbg Preview. Microsoft Docs [online]. Redmond, WA: Microsoft Press, 2022, 03/10/2022 [cit. 2022-08-02]. Dostupné z: https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/debugging-using-windbg-preview*

20. *GEREND, Jason. Autochk. Microsoft Docs [online]. Redmond, WA: Microsoft Press, 2021, 03/03/2021 [cit. 2022-08-10]. Dostupné z: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/autochk*

21. *RUSSINOVICH, Mark a Stephen PETERS. Portmon for Windows v3.03. Microsoft Docs [online]. Redmond, WA: Microsoft Press, 2022, January 12, 2012 [cit. 2022-08-04]. Dostupné z: https://docs.microsoft.com/en-us/sysinternals/downloads/portmon*

22. @Sysinternals. Twitter.com [online]. Redmond, WA: Mark Russinovich, 2008 [cit. 2022-06-27]. Dostupné z: https://twitter.com/Sysinternals

23. Sysinternals Blog [online]. Redmond, WA: Microsoft, 2019 [cit. 2022-06-27]. Dostupné z: https://techcommunity.microsoft.com/t5/sysinternals-blog/bg-p/Sysinternals-Blog

8 Přílohy

1. Přiloženo celé znění End User License Agreement pro nástroje Sysinternals.

SYSINTERNALS SOFTWARE LICENSE TERMS

These license terms are an agreement between Sysinternals (a wholly owned subsidiary of Microsoft Corporation) and you. Please read them. They apply to the software you are downloading from Systinternals.com, which includes the media on which you received it, if any. The terms also apply to any Sysinternals

- updates,
- supplements,
- Internet-based services, and
- support services

for this software, unless other terms accompany those items. If so, those terms apply.

BY USING THE SOFTWARE, YOU ACCEPT THESE TERMS. IF YOU DO NOT ACCEPT THEM, DO NOT USE THE SOFTWARE.

If you comply with these license terms, you have the rights below.

1. **INSTALLATION AND USE RIGHTS.** You may install and use any number of copies of the software on your devices.

2. **SCOPE OF LICENSE.** The software is licensed, not sold. This agreement only gives you some rights to use the software. Sysinternals reserves all other rights. Unless applicable law gives you more rights despite this limitation, you may use the software only as expressly permitted in this agreement. In doing so, you must comply with any technical limitations in the software that only allow you to use it in certain ways. You may not

   - work around any technical limitations in the binary versions of the software;
   - reverse engineer, decompile or disassemble the binary versions of the software, except and only to the extent that applicable law expressly permits, despite this limitation;
   - make more copies of the software than specified in this agreement or allowed by applicable law, despite this limitation;
   - publish the software for others to copy;
   - rent, lease or lend the software;
   - transfer the software or this agreement to any third party; or
   - use the software for commercial software hosting services.

3. **SENSITIVE INFORMATION.** Please be aware that, similar to other debug tools that capture "process state" information, files saved by Sysinternals tools may include personally identifiable or other sensitive information (such as usernames, passwords, paths to files accessed, and paths to registry accessed). By using this software, you acknowledge that you are aware of this and take sole responsibility for any personally identifiable or other sensitive information provided to Microsoft or any other party through your use of the software.

. **DOCUMENTATION.** Any person that has valid access to your computer or internal network may copy and use the documentation for your internal, reference purposes.

6. **EXPORT RESTRICTIONS.** The software is subject to United States export laws and regulations. You must comply with all domestic and international export laws and regulations that apply to the software. These laws include restrictions on destinations, end users and end use. For additional information, see www.microsoft.com/exporting <<http://www.microsoft.com/exporting>>.

7. **SUPPORT SERVICES.** Because this software is "as is, " we may not provide support services for it.

8. **ENTIRE AGREEMENT.** This agreement, and the terms for supplements, updates, Internet-based services and support services that you use, are the entire agreement for the software and support services.

9. **APPLICABLE LAW.**

   a. **United States.** If you acquired the software in the United States, Washington state law governs the interpretation of this agreement and applies to claims for breach of it, regardless of conflict of laws principles. The laws of the state where you live govern all other claims, including claims under state consumer protection laws, unfair competition laws, and in tort.

   b. **Outside the United States.** If you acquired the software in any other country, the laws of that country apply.

10. **LEGAL EFFECT.** This agreement describes certain legal rights. You may have other rights under the laws of your country. You may also have rights with respect to the party from whom you acquired the software. This agreement does not change your rights under the laws of your country if the laws of your country do not permit it to do so.

11. **DISCLAIMER OF WARRANTY. THE SOFTWARE IS LICENSED "AS - IS." YOU BEAR THE RISK OF USING IT. SYSINTERNALS GIVES NO EXPRESS WARRANTIES, GUARANTEES OR CONDITIONS. YOU MAY HAVE ADDITIONAL CONSUMER RIGHTS UNDER YOUR LOCAL LAWS WHICH THIS AGREEMENT CANNOT CHANGE. TO THE EXTENT PERMITTED UNDER YOUR LOCAL LAWS, SYSINTERNALS EXCLUDES THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.**

12. **LIMITATION ON AND EXCLUSION OF REMEDIES AND DAMAGES. YOU CAN RECOVER FROM SYSINTERNALS AND ITS SUPPLIERS ONLY DIRECT DAMAGES UP TO U.S. $5.00. YOU CANNOT RECOVER ANY OTHER DAMAGES, INCLUDING CONSEQUENTIAL, LOST PROFITS, SPECIAL, INDIRECT OR INCIDENTAL DAMAGES.**

    This limitation applies to

    - anything related to the software, services, content (including code) on third party Internet sites, or third party programs; and

    - claims for breach of contract, breach of warranty, guarantee or condition, strict liability, negligence, or other tort to the extent permitted by applicable law.

    It also applies even if Sysinternals knew or should have known about the possibility of the damages. The above limitation or exclusion may not apply to you because your country may not

allow the exclusion or limitation of incidental, consequential or other damages.

Please note: As this software is distributed in Quebec, Canada, some of the clauses in this agreement are provided below in French.

Remarque : Ce logiciel étant distribué au Québec, Canada, certaines des clauses dans ce contrat sont fournies ci-dessous en français.

EXONÉRATION DE GARANTIE. Le logiciel visé par une licence est offert « tel quel ». Toute utilisation de ce logiciel est à votre seule risque et péril. Sysinternals n'accorde aucune autre garantie expresse. Vous pouvez bénéficier de droits additionnels en vertu du droit local sur la protection dues consommateurs, que ce contrat ne peut modifier. La ou elles sont permises par le droit locale, les garanties implicites de qualité marchande, d'adéquation à un usage particulier et d'absence de contrefaçon sont exclues.

LIMITATION DES DOMMAGES-INTÉRÊTS ET EXCLUSION DE RESPONSABILITÉ POUR LES DOMMAGES. Vous pouvez obtenir de Sysinternals et de ses fournisseurs une indemnisation en cas de dommages directs uniquement à hauteur de 5,00 $ US. Vous ne pouvez prétendre à aucune indemnisation pour les autres dommages, y compris les dommages spéciaux, indirects ou accessoires et pertes de bénéfices.

Cette limitation concerne :

- tout ce qui est relié au logiciel, aux services ou au contenu (y compris le code) figurant sur des sites Internet tiers ou dans des programmes tiers ; et
- les réclamations au titre de violation de contrat ou de garantie, ou au titre de responsabilité stricte, de négligence ou d'une autre faute dans la limite autorisée par la loi en vigueur.

Elle s'applique également, même si Sysinternals connaissait ou devrait connaître l'éventualité d'un tel dommage. Si votre pays n'autorise pas l'exclusion ou la limitation de responsabilité pour les dommages indirects, accessoires ou de quelque nature que ce soit, il se peut que la limitation ou l'exclusion ci-dessus ne s'appliquera pas à votre égard.

EFFET JURIDIQUE. Le présent contrat décrit certains droits juridiques. Vous pourriez avoir d'autres droits prévus par les lois de votre pays. Le présent contrat ne modifie pas les droits que vous confèrent les lois de votre pays si celles-ci ne le permettent pas.

Zadání bakalářské práce

Autor:	**Martin Vahala**
Studium:	I1900270
Studijní program:	B1802 Aplikovaná informatika
Studijní obor:	Aplikovaná informatika
Název bakalářské práce:	**Využití nástrojů Windows Sysinternals pro efektivní analýzu operačního systému**
Název bakalářské práce AJ:	Using Windows Sysinternals for efficient analysis of the operating system

Cíl, metody, literatura, předpoklady:

Cílem práce je podrobně popsat vybrané nástroje ze skupiny Windows Sysinternals pro analýzu operačního systému a vytvoření sady praktických úloh na jejich využití.

V teoretické části práce autor popíše možnosti vybraných nástrojů Sysinternals System Information Utilities, Sysinternals Process Utilities a Sysinternals Security Utilities.

V praktické části pak autor zpracuje sadu úloh pro využití vybraných nástrojů Sysinternals System Information Utilities, Sysinternals Process Utilities a Sysinternals Security Utilities.

Zadávající pracoviště:	Katedra informačních technologií, Fakulta informatiky a managementu
Vedoucí práce:	Mgr. Josef Horálek, Ph.D.

Datum zadání závěrečné práce: 12.1.2021