

Рев'ю на тему: «АНАЛІЗ ФАЙЛОВИХ ОБ'ЄКТІВ ОПЕРАЦІЙНОЇ СИСТЕМИ WINDOWS 10 ДЛЯ ОЧИЩЕННЯ Й ОПТИМІЗАЦІЇ ПРОСТОРУ СИСТЕМНОГО РОЗДІЛУ»

Вступ

Наукова публікація авторів Булатецького В.В., Булатецької Л.В., Гришанович Т.О. «Аналіз файлових об'єктів операційної системи Windows 10 для очищення й оптимізації простору системного розділу» присвячена питанню ефективного використання системного розділу сучасної операційної системи Windows 10. Автори досліджують накопичення надлишкової, а інколи надмірної інформації в системних папках, причини заповнення дискового простору, засоби очищення для оптимізації використання ресурсу. Одним з питань роботи є вивчення структури файлових об'єктів ОС, які мають найбільший вплив на споживання дискового простору, а також обґрунтування ефективних методів його оптимізації за допомогою утиліт, як вбудованих так і сторонніх.

Основною метою дослідження є визначення об'єктів, що призводять до перевантаження диска, аналіз інструментів їх очищення та формування практичних рекомендацій щодо оптимізації системного середовища.

Методологія

У дослідженні застосовано:

- ❖ аналітичне дослідження структури Windows 10;
- ❖ систематизацію каталогів і файлів, що впливають на об'єм системного розділу;
- ❖ дослідження функціоналу інструментів очищення;
- ❖ порівняльний аналіз програмних рішень для оптимізації системного середовища.

В дослідженні автори базувалися на технічному аналізі вмісту системних каталогів (%SystemRoot%, WinSxS, Installer, DriverStore), типах даних у реєстрі ОС, засобах доступу до командного рядка, інтерпретаторів, механізмів запуску з підвищеними правами, розглядають rundll32 для автоматизації завдань. Авторами зроблено таблицю для порівняння функціоналу утиліт DISM, PatchCleaner, WiseCleaner, та DISM++, де вони позначили їх можливості(+ та -).

Результати

Отже, дослідження показують, що системний розділ Windows 10 може містити велику кількість файлових об'єктів, які споживають простір, але не є критичними для поточного функціонування системи. Найбільш важливими серед них виявилися:

- ❖ WinSxS - сховище оновлень, резервних копій і жорстких посилань;
- ❖ DriverStore/FileRepository - джерело накопичення старих версій драйверів;
- ❖ Installer - архів інсталяційних пакетів;
- ❖ Системні файли гібернації (hiberfil.sys)
- ❖ Файли віртуальної пам'яті (pagefile.sys, swapfile.sys);
- ❖ %SystemRoot%\System32\DllCache - копія системних файлів(резерв);
- ❖ %SystemDrive%\System Volume Information - сховище баз індексації, тінювих копій, точок відновлення.

Згадано про архітектурні особливості ОС, пов'язані з розміткою дисків, багатороздільну структуру, реалізацію інструментів через командний рядок (cmd, bat, PowerShell), механізм редагування реєстру через regedit, .reg - файли та reg - команди.

Графічні засоби, наприклад, cleanmgr(очищення диску, більш старий варіант) або Storage Sense(диспетчер пам'яті - автоматизований сучасний варіант) також не повністю можуть очистити простір для оптимізації. У зв'язку з чим згадуються сторонні засоби з ширшим функціоналом.

Ключові інсайти

👉 Архітектура Windows 10 включає об'єкти, що приховано нарощують споживання системного диску.

Системні файли (WinSxS, Installer, System Volume Information) мають службове призначення, але поступово накопичують об'єми, не очищуючись автоматично.

👉 Стандартні утиліти мають обмежений доступ до функціоналу системної оптимізації.

Вбудовані інструменти, мають спрощений набір функцій і не підтримують автоматизацію або роботу з об'єктами реєстру. Тому бажано знання PowerShell, CMD, та розуміння .bat - файлів.

☝ Структура реєстру - критичне джерело налаштувань, вплив на який може оптимізувати роботу системи.

Windows, а саме його реєстр - це база даних, що дозволяє налаштувати політики, служби, роботу драйверів. Розуміючи роботу .reg - файлів або користуючись командним рядком можна контролювати ОС без сторонніх утиліт.

☝ Жоден існуючий програмний продукт не забезпечує повного охоплення задач оптимізації.

Дивлячись на таблицю представлену у роботі, розумієш, що треба комбінувати продукти для продуктивної роботи системи.

Висновок

Дослідження у статті «Аналіз файлових об'єктів операційної системи Windows 10 для очищення й оптимізації простору системного розділу», окреслює проблеми, пов'язані з накопиченням файлів, архітектурною спадковістю, обмеженнями стандартних утиліт. Демонструє практичні шляхи розв'язання через скрипти, командні засоби, сторонні утиліти.

Робота має цінність для адміністраторів та можливо майбутніх інженерів служби підтримки




Можна зробити висновок, що розуміння оптимізації системного простору у Windows допомагає зрозуміти внутрішню логіку ОС та якісно керувати її ресурсами.

Додатково, у зв'язку з бажанням зрозуміти зв'язок отриманої інформації з напрямком кібербезпеки.





Зрозуміло, що у всіх цих файлах, сховищах та інструментах можливі недоліки. Ось як можна скористуватись недоліками, у випадку невірного налаштування, або не відслідковування стану згаданих файлів та сховищ, про які згадувалось у розділі «Результати»:

WinSxS

Атаки:




-  DLL Search Order Hijacking
-  Binary Planting
-  Shadow Copy Evasion

Інструменти:





-  PowerShell - для виявлення нестандартних посилань та аналізу доступу
-  CrackMapExec - перевірка доступу до файлів та прав
-  ProcMon - моніторинг завантаження DLL
-  Autoruns - аналіз автозапуску DLL

DriverStore / FileRepository

Атаки:

-  Driver DLL Hijacking
-  Malicious Driver Injection
-  Privilege Escalation через Signed Driver Abuse

Інструменти:

-  PowerView - для збору інформації про драйвери та привілеї
-  SharpHound - виявлення шляхів ескалації через драйвери
-  Sigcheck - перевірка цифрових підписів
-  WinDbg - аналіз нестабільних драйверів

Installer(%SystemRoot%\Installer)

Атаки:

- 🚩 Installer Hijacking
- 🚩 MSI Replay Persistence
- 🚩 Orphaned Installer Exploitation

Інструменти:

- 👉 PowerShell - для сканування вмісту Installer
- 👉 BloodHound - непрямий аналіз доступу до Installer через ACL
- 👉 PatchCleaner - виявлення сирітських MSI
- 👉 AccessChk - перевірка прав доступу

hiberfil.sys/ pagefile.sys/ swapfile.sys

Атаки:

- 🚩 Memory Dump Credential Harvesting
- 🚩 Post-Hibernation Extraction
- 🚩 Swap File Triage

Інструменти:

- 👉 Mimikatz - витяг облікових даних з пам'яті
- 👉 Impacket - для взаємодії з пам'яттю через SMB
- 👉 Volatility - аналіз дамів пам'яті
- 👉 DumpIt - створення дампу

%SystemRoot%\System32\DllCache

Атаки:

- 🚩 DllCache Poisoning

- 🚩 Forced Restoration of Malicious DLLs
- 🚩 Anti-Forensic File Replacement

Інструменти:

- 👉 CrackMapExec - перевірка доступу до системних DLL
- 👉 PowerShell - для порівняння хешів та вмісту
- 👉 Sigcheck - перевірка хешів
- 👉 FCIV - контроль цілісності

%SystemDrive%\System Volume Information

Атаки:

- 🚩 VSS Persistence
- 🚩 Shadow Copy Hijacking
- 🚩 Restore Point Injection

Інструменти:

- 👉 PowerShell - для аудиту VSS
- 👉 SharpHound - непрямий аналіз доступу через ACL
- 👉 vssadmin - керування точками відновлення
- 👉 FTK Imager - форензика копій

.bat файли

Атаки:

- 🚩 Command Injection
- 🚩 Scripted Privilege Abuse
- 🚩 Scheduled Task Persistence

Інструменти:

- 👉 PowerShell - для аналізу запуску .bat
- 👉 gpclient - перевірка прав доступу
- 👉 Sysmon - логування запусків

regedit / реєстр Windows

Атаки:

- 🚩 Run Key Persistence
- 🚩 Shell Extension Hijacking
- 🚩 Registry Reflection Abuse

Інструменти:

- PowerShell - для читання та зміни ключів
- SharpHound - аналіз привілеїв через реєстр
- RegRipper - глибокий аналіз
- RECmd - CLI-доступ до реєстру

П.С. Як бачимо інструментів дуже багато, що ще раз нагадує нам про необхідність розуміння логіки побудови системи Windows, для оптимізації її роботи.

https://www.google.com/search?q=Cleanmgr+%D0%B0%D0%B1%D0%BE+Storage+Sense&oq=Cleanmgr+%D0%B0%D0%B1%D0%BE+Storage+Sense+&gs_lcrp=EgZjaHJvbWUyBggAEEUYOTIHCAEQIRigATIHCAlQIRigATIHCAMQIRigATIHCACQIRigAdIBCTIwODFqMGoxNagCCLACAfEFbe4s76scwfxBW3uLO-rHHsH&sourceid=chrome&ie=UTF-8
