УДК 004.451.9:004.42

ВИКОРИСТАННЯ СЦЕНАРІЇВ POWERSHELL ДЛЯ НАЛАШТУВАННЯ РОБОЧИХ СТАНЦІЙ

Бровко Д. Д.
my4enie777@gmail.com
Черкаський державний фаховий бізнес коледж
Фальченко Н. Г.
м.Черкаси, Україна

У сучасній системній інженерії та управлінні IT-інфраструктурою особливої ваги набуває концепція «інфраструктура як код» (Infrastructure as Code). Автоматизація розгортання та налаштування робочих станцій дозволяє суттєво скоротити час на введення обладнання в експлуатацію, мінімізувати людський фактор та забезпечити ідемпотентність – стан, при якому повторне виконання скрипта приводить до одного й того самого прогнозованого результату. Використання сценаріїв PowerShell є одним із найбільш ефективних методів керування середовищем Windows завдяки глибокій інтеграції з операційною системою та доступу до об’єктної моделі .NET [1].

Основні завдання розробки включають: аналіз існуючих методів розгортання (MDT, SCCM, ручне налаштування), проєктування модульної архітектури скриптів, реалізацію механізмів автоматичного встановлення прикладного ПЗ, конфігурування параметрів безпеки та мережевих інтерфейсів, а також створення системи звітності про стан завершення процесів [2].

Система реалізована на базі мови сценаріїв PowerShell версії 5.1/7.x. Взаємодія з компонентами ОС базується на використанні WMI (Windows Management Instrumentation) та CIM-сесій, що дозволяє отримувати вичерпну інформацію про апаратне забезпечення та керувати системними службами. Головною перевагою такого підходу є відсутність потреби у сторонніх агентах (agentless-підхід) та можливість гнучкого налаштування реєстру, профілів користувачів і системних політик. Для управління пакунками ПЗ інтегровано використання менеджерів пакетів, таких як Winget та Chocolatey, що забезпечує

автоматичне завантаження та «тихе» встановлення актуальних версій програм [3].

Особлива увага приділяється забезпеченню безпеки виконання сценаріїв. Реалізовано перевірку політик виконання (Execution Policy), обробку виняткових ситуацій через блоки Try-Catch-Finally та логування кожного етапу розгортання у текстові файли або події Windows (Event Log). Це критично важливо для діагностики помилок у великих корпоративних мережах. Крім того, розроблено механізм ідентифікації обладнання, що дозволяє скрипту адаптивно встановлювати специфічні драйвери залежно від моделі робочої станції [4].

Функціонал розробленого комплексу сценаріїв включає:

- модуль первинної ініціалізації ОС (регіональні стандарти, назва ПК);
- блок керування мережевими налаштуваннями та приєднанням до домену Active Directory;
- систему автоматичного встановлення базового та спеціалізованого ПЗ;
- модуль оптимізації продуктивності та вимкнення небажаних телеметричних служб;
- інтерфейс фінальної перевірки цілісності конфігурації. Усі компоненти побудовані за принципом модульності, що дозволяє легко додавати нові функції без зміни ядра системи [5].

Логіка роботи побудована на послідовному виконанні фаз: ініціалізація, конфігурування, верифікація. Для прискорення процесу впроваджено методи паралельних обчислень (PowerShell Jobs/Runspaces) при встановленні незалежних компонентів ПЗ. Для інформування системного адміністратора про хід роботи додано графічні прогрес-бари або консольну індикацію статусів.

Додатково реалізовано систему аудиту, яка після завершення роботи скриптів формує фінальний звіт (HTML або CSV). Це дозволяє миттєво оцінити, чи всі параметри було застосовано коректно, та зафіксувати час, витрачений на підготовку однієї станції. Енергоаудит та моніторинг ресурсів під час виконання скриптів підтвердили низьке навантаження на процесор та пам’ять.

Тестування проводиться на віртуальних машинах та фізичних робочих станціях з різними конфігураціями заліза. Результати показали, що автоматизований підхід скорочує час налаштування робочого місця на 70–85% порівняно з ручним методом. Пристрій демонстрував високу стабільність навіть за умов нестабільного мережевого з’єднання завдяки механізмам повторних запитів (Retry logic).

У перспективі планується інтеграція розроблених рішень із хмарними сервісами (Microsoft Intune/Azure Autopilot), додавання підтримки PowerShell Core для кросплатформного керування та розробка веб-інтерфейсу для моніторингу статусу розгортання в реальному часі.

Таким чином, розроблена система автоматизації на основі PowerShell є потужним інструментом для сучасного системного адміністратора, що поєднує гнучкість, масштабованість та високу надійність. Реалізовані технічні рішення дозволяють забезпечити уніфікацію робочих місць та значне підвищення ефективності ІТ-відділу.

Список використаних джерел:

1. Бернацький А. В. Автоматизація адміністрування Windows за допомогою PowerShell. Технічні науки та технології. 2021. № 2 (24). С. 112–118.
2. Шевченко О. М., Петренко В. І. Порівняльний аналіз інструментів розгортання робочих станцій у корпоративних мережах. Системи обробки інформації. 2023. Вип. 3 (172). С. 45–52.
3. Коваленко С. С. Використання менеджерів пакетів для автоматизації встановлення ПЗ. Комп’ютерно-інтегровані технології. 2022. Т. 48. С. 89–94.
4. Мельник Д. Р. Забезпечення безпеки та цілісності сценаріїв автоматизації в ОС Windows. Кібербезпека: освіта, наука, техніка. 2024. № 1 (21). С. 130–137.

5. Сидоренко В. А. Оптимізація IT-інфраструктури підприємства через впровадження методів Infrastructure as Code. Цифрова економіка та системи управління. 2024. Вип. 12. С. 15–22.

УДК 004.8:004.415

АРХІТЕКТУРНІ ПІДХОДИ ТА ІНЖЕНЕРНІ ПРАКТИКИ РОЗРОБКИ ДЕЦЕНТРАЛІЗОВАНИХ ВЕБ-ЗАСТОСУНКІВ НА БАЗІ БЛОКЧЕЙНУ

Скубій Є.В.
yevgeniyaskubiy@gmail.com
Черкаський державний фаховий бізнес-коледж
Дмитрюк В.В.
м.Черкаси, Україна

Розвиток Web 3.0 потребує створення децентралізованих застосунків (dApps), які забезпечують безпечну взаємодію з блокчейном без передачі приватних ключів. Перехід від моделі «клієнт-сервер» до децентралізованих систем вимагає нових підходів до архітектури, безпеки та оптимізації витрат (газ-комісій). Через складність управління станом транзакцій та специфіку UX, систематизація інженерних практик розробки dApps є критично важливою для забезпечення надійності та масштабованості сучасних вебсервісів.

Метою дослідження є аналіз сучасних архітектурних підходів та інженерних практик розробки децентралізованих вебзастосунків на базі блокчейну, систематизація вимог до їхнього проєктування та вивчення технологічних рішень для забезпечення безпеки, масштабованості й зручності використання [1].

Децентралізований застосунок базується на моделі «клієнт-блокчейн», що замінює класичну схему «клієнт-сервер». Архітектура включає три рівні: на рівні блокчейну смартконтракти (Solidity, Rust) реалізують логіку в детермінованому та невідворотному вигляді; проміжний рівень (JSON-RPC шлюзи та API) забезпечує зв'язок фронтенду з мережею; на рівні фронтенду інтерфейси взаємодіють із гаманцями (як MetaMask) для безпечного підписання транзакцій.