Методика користування технологіями контейнеризації під час проведення судових комп'ютерно-технічних експертиз програмного забезпечення вебсайтів

Руслан Мірошник
судовий експерт, лабораторія досліджень об'єктів інформаційних технологій, Науково-дослідний центр судової експертизи у сфері інформаційних технологій та інтелектуальної власності Міністерства юстиції України, м. Київ, Україна, ORCID: https://orcid.org/0009-0004-8504-0489, e-mail: ruslan_miroshnik@ukr.net

Володимир Гомон
завідувач лабораторії досліджень об'єктів інформаційних технологій, Науково-дослідний центр судової експертизи у сфері інформаційних технологій та інтелектуальної власності Міністерства юстиції України, м. Київ, Україна, ORCID: https://orcid.org/0000-0001-6326-9590, e-mail: gomon1995@gmail.com

Дмитро Хівренко
судовий експерт, лабораторія досліджень об'єктів інформаційних технологій, Науково-дослідний центр судової експертизи у сфері інформаційних технологій та інтелектуальної власності Міністерства юстиції України, м. Київ, Україна, ORCID: https://orcid.org/0009-0009-9718-5534, e-mail: dimankhivrenko@gmail.com

Розглянуто актуальну проблему дослідження вебсайтів, наданих на експертизу у формі вихідного коду та структур даних (так званої сирої структури). Обґрунтовано користування технологією Docker Compose як ефективним інструментом для створення ізольованого, відтворюваного й безпечного експертного середовища. Описано покроковий алгоритм розгортання досліджуваних об'єктів, що дає змогу уникнути модифікації вихідних даних. Проаналізовано переваги контейнеризації порівняно з традиційними методами віртуалізації (зокрема, у контексті автоматизації налаштування залежностей і мінімізації ризиків для автоматизованого робочого місця експерта).

Ключові слова: *судова комп'ютерно-технічна експертиза; Docker; Docker Compose; вихідний код; вебсайт; контейнеризація; цифрові докази.*

Methodology for the use of Containerisation Technologies in the Conduct jf Forensic Computer and Technical Examinations of Website Software

Ruslan Miroshnyk, Volodymyr Homon, Dmytro Khivrenko

The article addresses the pressing issue of investigating websites submitted for forensic examination in the form of source code and data structures (the so-called «raw structure»). The authors justifies the use of Docker Compose technology as an efficient tool for creating an isolated, reproducible, and secure forensic environment. A step-by-step algorithm for deploying investigated objects is described, which prevents modification of original data and ensures the integrity of the forensic experiment. The advantages of containerization over traditional virtualization methods are analyzed, particularly in the context of automating dependency configuration and minimizing risks to the forensic workstation.

Keywords: *forensic computer examination; Docker; Docker Compose; source code; website; containerization; digital evidence.*

Актуальність теми. Стрімкий розвиток вебтехнологій і цифровізація суспільних відносин призвели до того, що об'єктами судових комп'ютерно-технічних експертиз та експертиз електронних комунікацій дедалі частіше стають складні вебдодатки й сервіси. У процесі досудового розслідування експерти стикаються з проблемою дослідження об'єктів, наданих не у вигляді функціонуючого ресурсу в інтернеті, а як сукупність вихідного коду та дампів баз даних на цифрових носіях. Така «сира структура» вебсайту потребує від експерта не лише статичного аналізу тексту програмного коду, а й повного відтворення працездатного стану об'єкта для дослідження реалізації функцій вебсайту, передбачених технічним завданням.

Аналіз останніх досліджень. Питання дослідження вебсайтів розглядали багато українських учених-криміналістів, проте

традиційні підходи, що базуються на користуванні локальними серверами, мають суттєві недоліки: відсутність повної ізоляції, складність у відтворенні специфічних залежностей і ризик модифікації вихідних даних.

Мета статті полягає в обґрунтуванні та розробленні методичного підходу до проведення судової комп'ютерно-технічної експертизи вебсайтів на предмет їх відповідності функціональним вимогам технічного завдання. Основну увагу приділено застосуванню технології *Docker Compose* як засобу створення ідентичного й ізольованого середовища для динамічної верифікації програмного продукту, що дає змогу експертові надати об'єктивну оцінку повноти реалізації функцій, передбачених умовами договору, мінімізуючи водночас ризики модифікації об'єкта дослідження.

Викладення основного матеріалу. Процес експертного дослідження за допомогою технології *Docker Compose* поділяється на три основні етапи: підготовчий (статичний), операційний (динамічний) та етап верифікації.

На етапі статичної декомпозиції об'єкта дослідження і технічного завдання експерт проводить порівняльний аналіз документації та структури файлів. Дослідження технічного завдання починається з аналізу функціональних вимог. Експерт формує реєстр функціональних точок, які система повинна виконувати для задоволення потреб користувача. Кожну таку точку згодом перевіряють в ізольованому середовищі *Docker* на предмет фактичного існування та коректності роботи.

Для повноцінного дослідження експерт групує вимоги за функціональними блоками. Далі наведено орієнтовний перелік точок, які найчастіше стають об'єктом перевірки:

1. *Блок ідентифікації та безпеки:*
   - *система реєстрації* — перевірка алгоритмів створення облікових записів (через *e-mail*, номер телефону або соціальні мережі *OAuth 2.0*);
   - *механізм автентифікації* — наявність процедур шифрування паролів (хешування за стандартами *Argon2*, *BCrypt*);
   - *відновлення доступу* — реалізація функцій «Забули пароль?» із застосуванням токенів з обмеженим терміном дії;
   - *рольова модель (RBAC)* — розмежування прав доступу (Користувач / Модератор / Адміністратор) і захист адміністративної панелі.
2. *Блок управління контентом (CMS-functional):*
   - *CRUD-операції* — можливість створення (*Create*), читання (*Read*), оновлення (*Update*) та видалення (*Delete*) публікацій або товарів;
   - *медіа-менеджмент* — наявність модулів для завантаження, стискання та зберігання зображень / відео на сервері;
   - *система тегування та категорій* — реалізація деревоподібної структури даних для навігації.
3. *Комерційний і фінансовий блоки:*
   - *кошик та обробка замовлень* — логіка додавання товарів, перерахунку кількості та збереження сесії користувача;
   - *інтеграція з платіжними шлюзами* — наявність коду для взаємодії з API (LiqPay, Stripe, PayPal) та обробка Callback-відповідей від банку;
   - *генерація інвойсів* — автоматичне створення рахунків у форматі PDF на основі даних замовлення.
4. *Комунікаційний та інтеграційний блоки:*
   - *форми зворотного зв'язку* — валідація введених даних на стороні сервера та відправка сповіщень через *SMTP/API* (*PHPMailer*, *SendGrid*);
   - *пошукові алгоритми* — реалізація повнотекстового пошуку з урахуванням фільтрів і сортування за ціною / рейтингом;
   - *інтеграція зі сторонніми сервісами* — робота з *API* карт (*Google*

Maps), служб доставки або *CRM*-систем;

5. *Службовий і системний блоки:*
   - *багатомовність (i18n)* — наявність механізмів перемикання мовних версій сайту через файли локалізації або бази даних;
   - *система логування* — фіксація критичних помилок і дій адміністраторів у системних журналах (*Logs*);
   - *SEO-оптимізація* — наявність полів для мета-тегів, генерація файлів sitemap.xm.

Після проведення статичного аналізу постає завдання перевести об'єкт із пасивного стану (набір файлів) у динамічний (працюючий застосунок). Для запуску «сирої» структури експерт розробляє архітектуру взаємодії сервісів. Основна перевага *Docker Compose* полягає в можливості описати складні зв'язки в декларативному вигляді. Формування середовища для дослідження за допомогою контейнеризації дає змогу створити контейнер, у якому об'єкт дослідження функціонує незалежно від операційної системи автоматизованого робочого місця експерта. Приклад конфігурації для проведення експертного дослідження зображено на рис. 1.

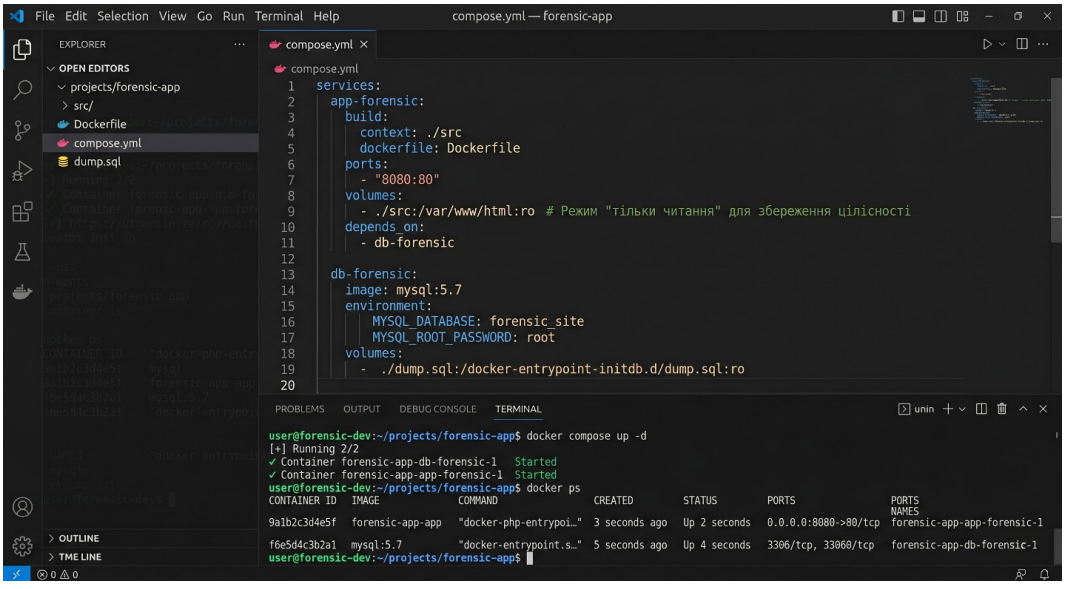

Рис. 1. Зображення конфігурації для проведення дослідження

Користування *Docker Compose* дає змогу експертові зафіксувати всі налаштування сервера в одному файлі. Основні параметри конфігурації для забезпечення експертного дослідження:

- директива *image* — користування конкретними версіями образів (наприклад, *php:7.4-apache* замість *latest*). Це гарантує, що сайт працюватиме саме в тих умовах, які були актуальними на момент розробки згідно з технічним завданням;
- механізм *Volumes* із прапорцем *:ro* (*Read-Only*) — монтування папки з вихідним кодом у режимі «тільки читання», унеможливлює випадкову зміну файлів об'єкта (наприклад, створення кешу, зміну конфігураційних файлів або впровадження скриптів) під час дослідження;
- директива *depends_on* — визначає чітку послідовність запуску сервісів, забезпечуючи готовність бази даних до моменту ініціалізації вебсайту.

На наступному етапі здійснюють активацію середовища та первинну перевірку цілісності. Після виконання команди *docker compose up -d* (*див. рис. 2*), експерт повинен

переконатися, що середовище розгорнуто коректно, і здійснити:

- перевірку статусів контейнерів — команда «*docker compose ps*» має показувати стан *Up* для всіх сервісів;
- аналіз логування ініціалізації — перегляд логування бази даних (*docker compose logs db*) на предмет успішного імпорту *SQL*-дампу без помилок синтаксису;
- верифікацію доступності — перехід за локальною адресою (наприклад, *http://localhost:8080*) для підтвердження того, що вебсервер коректно обробляє вхідні з'єднання (*див. рис. 3*).

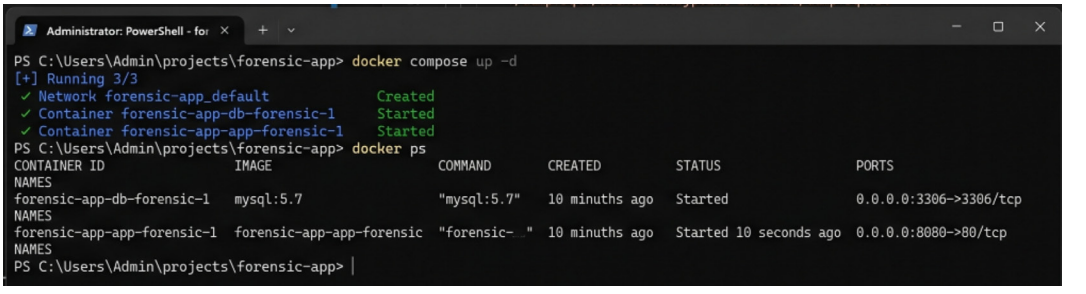

Рис. 2. Зображення виконання команди *docker compose up –d*

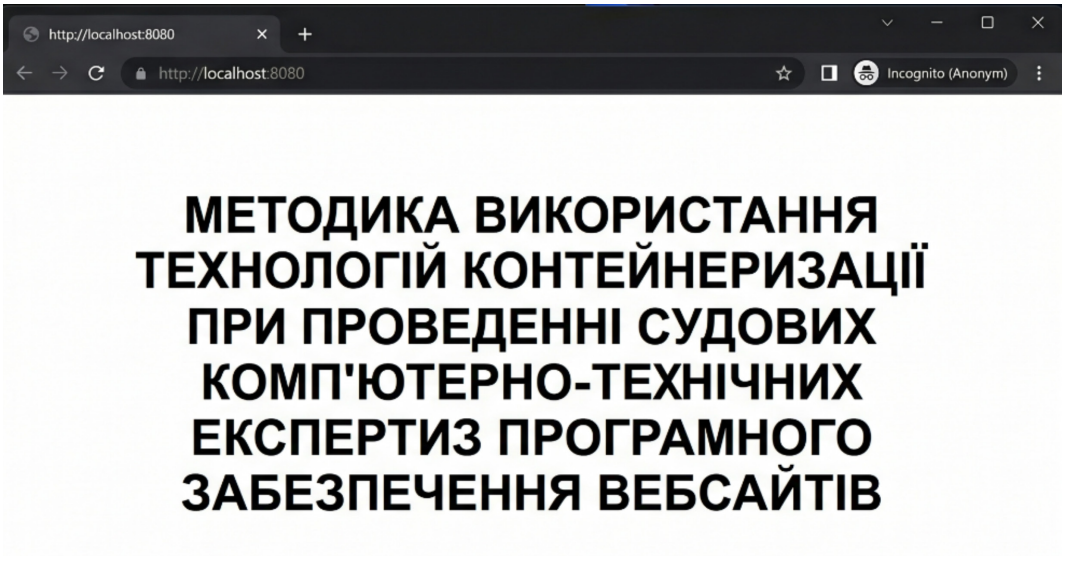

Рис. 3. Зображення верифікації доступності вебсервера

Наступний етап — динамічна верифікація, яка полягає в систематичному тестуванні кожної функціональної точки, виокремленої на етапі аналізу технічного завдання, за допомогою ініціації відповідних запитів до розгорнутого вебсайту й аналізу відповідей системи. Перевірка відповідності функціоналу вебсайту вимогам технічного завдання — це процес динамічної верифікації, де кожна функціональна вимога має бути підтверджена конкретним технічним станом програмного коду та його поведінкою в ізольованому робочому середовищі *Docker*. Процес верифікації поділяється на три ієрархічні рівні аналізу:

1) архітектурний рівень (наявність компонентів) — експерт перевіряє, чи існують у структурі сайту програмні модулі та бази даних, необхідні для виконання функцій;

2) логічний рівень (коректність алгоритмів) — на цьому етапі перевіряють, чи за кожним елементом інтерфейсу стоїть реальна бізнес-логіка, а не візуальна імітація.
3) результативний рівень (цілісність даних) — фінальна перевірка того, чи призводить виконання функції до очікуваної зміни стану системи.

За результатом дослідження експерт формує підсумкову таблицю відповідності функціоналу вебсайту вимогам технічного завдання, яка наочно демонструє повноту виконання функціональних вимог. Завершення етапу динамічної верифікації дає експертові змогу дійти однозначного висновку: функція «реалізована повністю», «реалізована частково (з помилками)» або «не реалізована зовсім (існує лише візуальна імітація)».

Висновки. У результаті проведеного дослідження обґрунтовано доцільність та ефективність застосування технології контейнеризації *Docker Compose* під час проведення судових комп'ютерно-технічних експертиз та експертиз електронних комунікацій вебсайтів. Розроблений методичний підхід дає змогу трансформувати процес дослідження «сирої структури» програмного продукту з пасивного спостереження за кодом в активний експертний експеримент. Запровадження цього методу в експертну практику сприяє значному скороченню часу на підготовку середовища дослідження та убезпечує автоматизоване робоче місце експерта під час роботи з неперевіреним програмним кодом.

Перелік джерел посилання

1. Інструкція про призначення та проведення судових експертиз та експертних досліджень : затв. наказ. Мін'юсту України від 08.10.1998 р. № 53/5 (зі змін. та допов.). URL: https://zakon.rada.gov.ua/laws/show/z0705-98#Text (дата звернення: 14.04.2026).
2. Климчук М. П., Комісарчук Ю. А., Марко С. І., Стецик Б. В. Судова комп'ютерно-технічна експертиза у кримінальному провадженні : навч. посіб. Львів, 2022. 112 с. URI: http://dspace.lvduvs.edu.ua/handle/1234567890/4399 (дата звернення: 14.04.2026).
3. Souppaya M., Morello J., Scarfone K. Application Container Security Guide / NIST Special Publication 800-190. Gaithersburg : NIST, 2017. 63 p. DOI: 10.6028/NIST.SP.800-190 (дата звернення: 14.04.2026).
4. Cito J., Ferme V., Gall H. C. Using Docker Containers to Improve Reproducibility in Software and Web Engineering Research. *Web Engineering. ICWE 2016. Lecture Notes in Computer Science.* Cham : Springer, 2016. Vol. 9671. P. 609–612. DOI: 10.1007/978-3-319-38791-8_58 (дата звернення: 14.04.2026).
5. Spiekermann D., Eggendorfer T., Keller J. A Study of Network Forensic Investigation in Docker Environments. *Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES 2019).* NY : ACM, 2019. Art. 44. P. 1–7. DOI: 10.1145/3339252.3340505 (дата звернення: 14.04.2026).