

Использование Kali Linux и Docker для безопасной и эффективной работы

 purpleschool.ru/knowledge-base/article/kali



23 июня 2025

Автор

Олег Марков

Введение

Kali Linux — это одна из самых популярных операционных систем для тестирования безопасности. Она предлагает множество встроенных инструментов для пентестинга, анализа сети и цифровой криминалистики. Docker, в свою очередь, предоставляет возможность изолированного выполнения приложений с помощью контейнерных технологий. Объединение этих инструментов позволяет создавать надежные и масштабируемые среды для тестирования и разработки. В этой статье мы разберемся, как использовать Kali Linux и Docker вместе, чтобы улучшить вашу рабочую среду и повысить её безопасность и эффективность.

Преимущества использования Kali Linux

Kali Linux разработана специально для тестирования безопасности и предлагает более 600 предустановленных инструментов для выполнения таких задач, как сканирование уязвимостей, атаки на беспроводные сети и многое другое. У этой операционной системы регулярные обновления, которые поддерживают актуальность её инструментов.

Возможности Docker

Docker — это платформа, которая позволяет разработчикам создавать, развертывать и запускать приложения в контейнерах. Контейнеры обеспечивают изоляцию процессов, что повышает безопасность и удобство работы. Благодаря Docker можно легко переносить приложение между разными системами, обеспечивая его стабильное выполнение без зависимости от окружения.

Установка и настройка Docker в Kali Linux

Установка Docker

Первым шагом в интеграции Docker с Kali Linux является его установка. Давайте начнем с добавления официального репозитория Docker и установки необходимых пакетов:

```
# Обновляем систему и устанавливаем требуемые пакеты
sudo apt-get update
sudo apt-get install -y apt-transport-https ca-certificates curl gnupg lsb-release

# Добавляем Docker GPG-ключ
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg

# Добавляем репозиторий Docker
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-
archive-keyring.gpg] https://download.docker.com/linux/debian $(lsb_release -cs)
stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Устанавливаем Docker
sudo apt-get update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io
```

Проверка установки

После установки Docker вы можете проверить, работает ли он, с помощью следующей команды:

```
# Проверяем версию Docker, чтобы убедиться в успешной установке
docker --version
```

Если установка прошла успешно, вы увидите версию Docker, установленную на вашей системе.

Настройка Docker

После установки Docker рекомендуется добавить пользователя в группу Docker, чтобы избегать использования **sudo** для каждой команды Docker:

```
# Добавляем пользователя в группу Docker
sudo usermod -aG docker $USER

# Применяем изменения, выйдя из текущей сессии и войдя снова
```

Эти шаги помогут сделать вашу работу с Docker более удобной и безопасной.

Использование Docker для запуска инструментов Kali Linux

Запуск контейнера с Kali Linux

Теперь, когда Docker установлен и настроен, давайте создадим контейнер с Kali Linux. Это позволит вам использовать все возможности Kali Linux в изолированной контейнеризированной среде:

```
# Загружаем образ Kali Linux
docker pull kalilinux/kali-rolling

# Запускаем контейнер в интерактивном режиме
docker run -it kalilinux/kali-rolling /bin/bash
```

Установка инструментов в контейнере

После запуска контейнера вы можете установить любые необходимые инструменты Kali Linux. Например, установка мета-пакета `kali-linux-top10`, который включает в себя 10 самых популярных инструментов тестирования безопасности:

```
# Устанавливаем мета-пакет с основными инструментами
apt-get update
apt-get install -y kali-linux-top10
```

Сохранение изменений в контейнере

Если вы установили новые пакеты или изменили настройки в контейнере, вы можете сохранить эти изменения, создав новый образ из измененного контейнера:

```
# Получаем ID контейнера
CONTAINER_ID=$(docker ps -l -q)

# Сохраняем изменения в новый образ
docker commit $CONTAINER_ID my-kali-image
```

Теперь вы можете использовать образ `my-kali-image` для создания новых контейнеров с сохраненными изменениями.

Автоматизация процессов с использованием Docker Compose

Создание файла Docker Compose

Docker Compose позволяет описывать многоконтейнерные приложения с помощью файла `docker-compose.yml`. Вот пример, как можно использовать Docker Compose для автоматизации создания контейнеров с Kali Linux:

```
version: '3.8'
services:
  kali:
    image: kalilinux/kali-rolling
    volumes:
      - ./data:/data
    command: /bin/bash
```

Запуск Docker Compose

Для запуска контейнеров с Kali Linux, описанных в файле `docker-compose.yml`, используйте следующую команду:

```
# Запуск всех сервисов, описанных в docker-compose.yml
docker-compose up
```

Этот подход делает запуск и управление контейнерами более удобным и позволяет интегрировать другие сервисы, которые могут понадобиться при тестировании или разработке.

Заключение

Итак, вы узнали, как использовать Kali Linux и Docker совместно для создания безопасной и эффективной среды тестирования и разработки. Использование Docker позволяет запускать приложения в изолированной среде, что повышает их безопасность, а также упрощает переносимость и масштабируемость. Kali Linux, в свою очередь, предоставляет все необходимые инструменты для тестирования безопасности и анализа. Такой метод работы подойдет как небольшим командам, работающим над пентестами, так и крупным компаниям, стремящимся обеспечить безопасность и стабильность своих приложений.