Отчёт по выполнению лабораторной работы №2

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Закрепить практические навыки устранения уязвимостей и защиты интеграционной платформы

2 Задание

Провести анализ уязвимостей, устранить их и последствия

3 Выполнение

3.1 Уязвимый узел Bitrix

Эксплуатация данной уязвимости позволяет удаленному нарушителю записать произвольные файлы в систему с помощью отправки специально сформированных сетевых пакетов.

Сначала заведём карточку с описание уязвимости, ее индикаторами и рекомендациями по устранению.

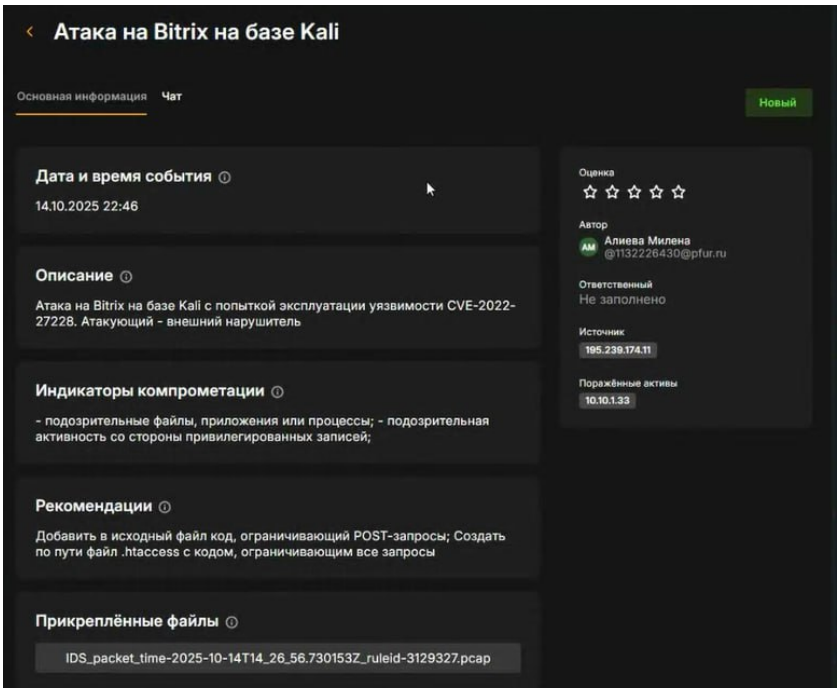

Рис. 3.1: Карточка уязвимости

Теперь начинаем устранять уязвимость. Для начала необходимо закрыть вектор для локального повышения привилегий, для этого удаляем SUID-бит у файла /var/www/html/apache_restart с помощью команды chmod –s /var/www/html/apache_restart и удаляем файл /var/www/html/apache_restart с помощью команды rm /var/www/html/apache_restart

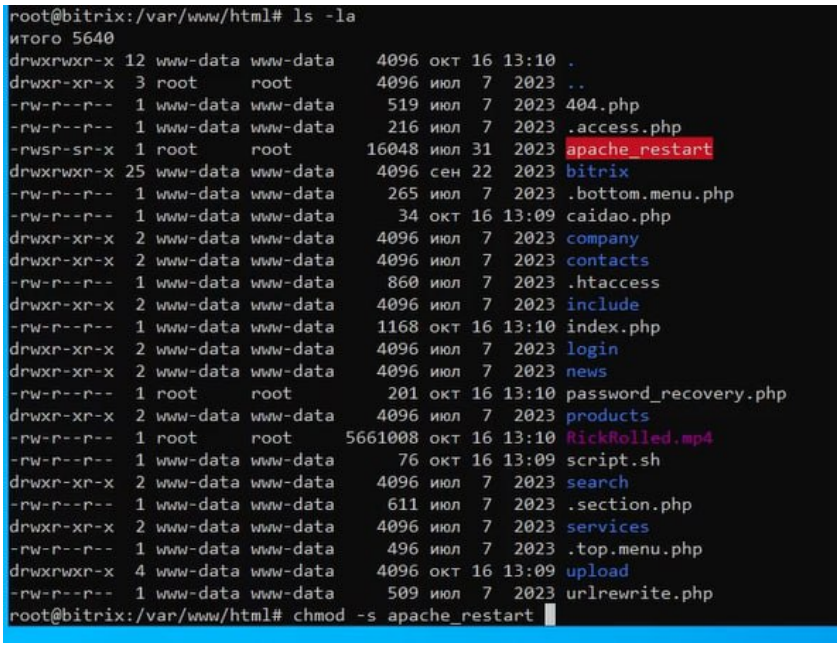

Рис. 3.2: Закрытие вектора для локального повышения привелегий

После закрытия локального повышения привилегий можно приступить к закрытию уязвимости CVE-2022-27228 несколькими способами. Для этого мы создали файл .htaccess, который отклоняет все запросы к директории vote.

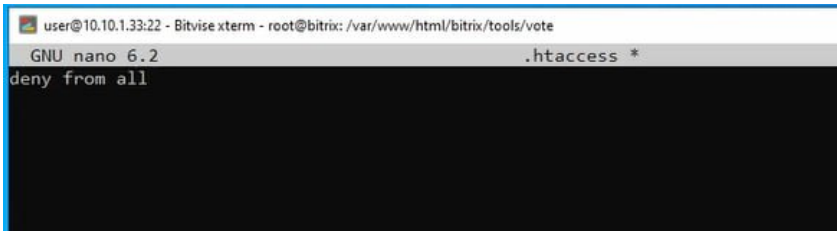

Рис. 3.3: Закрытие локального повышения привилегий

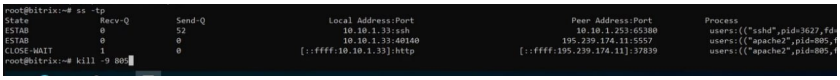

Рис. 3.4: Закрытие сессии

Теперь уязвимость устранена:

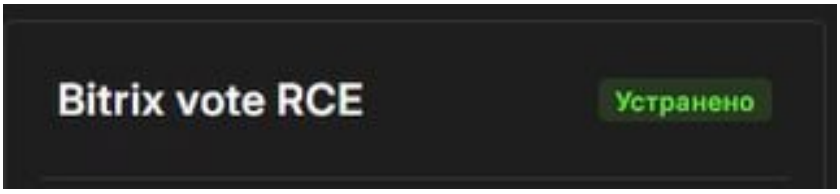

Рис. 3.5: Закрытие уязвимости

Затем нейтрализуем полезную нагрузку. В нашем случае полезная нагрузка меняет пароль от учетной записи администратора, в связи с чем невозможно получить доступ к панели администрирования. Если подключиться на сервер Bitrix по протоколу SSH, то в директории веб-сервера можно обнаружить скрипт password_recovery.php. Нам необходимо поменять в нём пароль

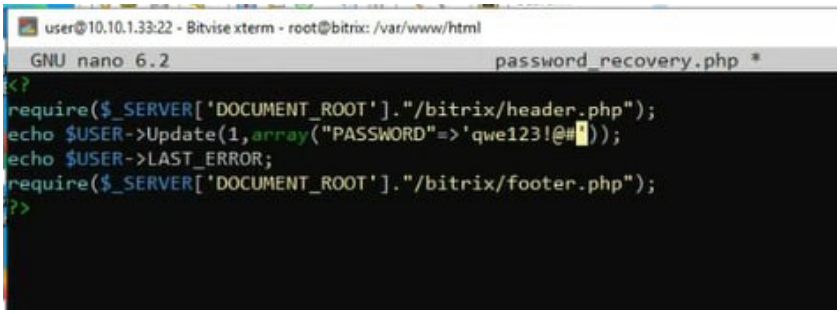

Рис. 3.6: Изменение пароля

Затем после восстановления доступа к панели администрирования можно приступить к восстановлению сайта после использования полезной нагрузки. В первую очередь удаляем все файлы в директории взломанного веб-сервера с помощью команды и файл резервной копии разархивируем в нужную директорию.

Рис. 3.7: Восстановление сайта

Теперь последствие устранено:

Рис. 3.8: Закрытие последствия

3.2 Уязвимый узел Gitlab

Используемый на платформе сервер GitLab версии 13.10.2 содержит критическую уязвимость CVE-2021-22204, которая позволяет получить RCE при загрузке определенных файлов в репозиторий. Уязвимость заключается в том, что при загрузке файлов с расширением JPG, jpeg, tiff, модуль GitLab Workhorse передает файлы в библиотеку ExifTool, которая удаляет из них метаданные.

Сначала заведём карточку с описание уязвимости, ее индикаторами и рекомендациями по устранению.

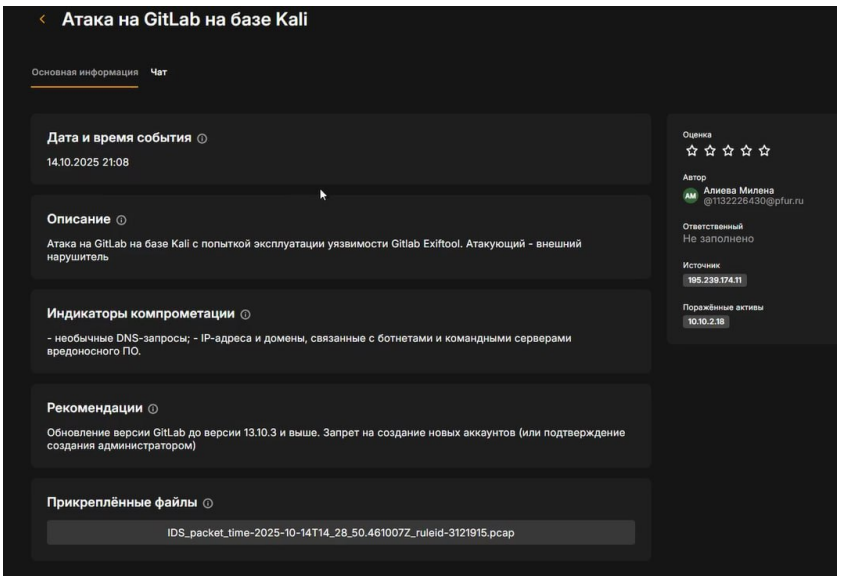

Рис. 3.9: Карточка уязвимости

Сначала изменим параметры регистрации новых пользователей, для этого перейдём на страницу авторизации Gitlab:

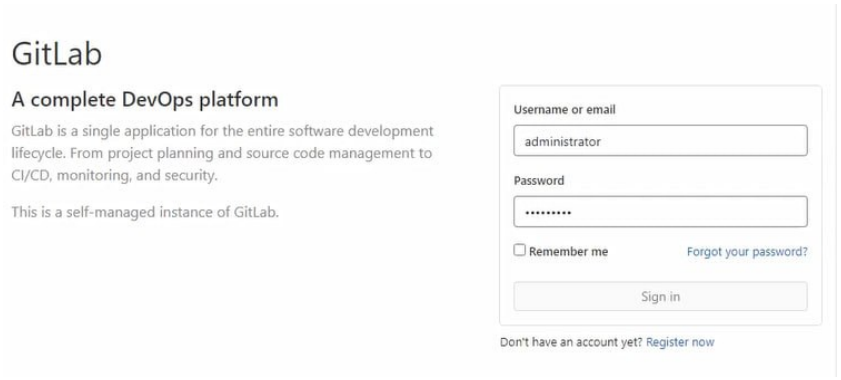

Рис. 3.10: Авторизация Gitlab

Далее переходим в Admin Area и ищем пункт Sign-up restrictions, расширяем его, позволяем добавление новых пользователей только с одобрения администратора

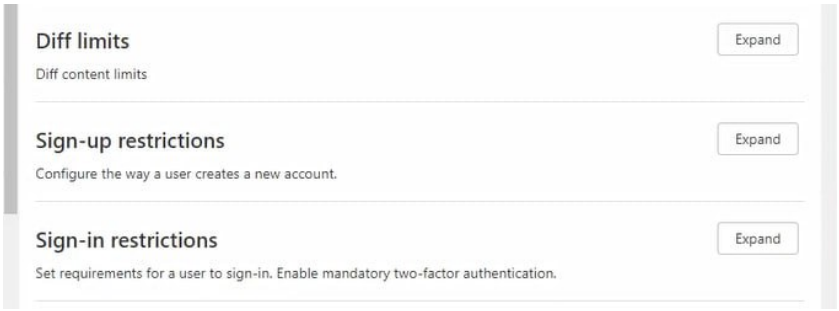

Рис. 3.11: Настройки Gitlab

После указанных действий нарушитель не сможет регистрировать новые учетные записи на сервере, но так как нарушитель уже проводил эксплуатацию ранее, то на сервере все еще существуют вредоносные учетные записи, их нужно удалить. В настройках переходим во вкладку Users, удаляем пользователя:

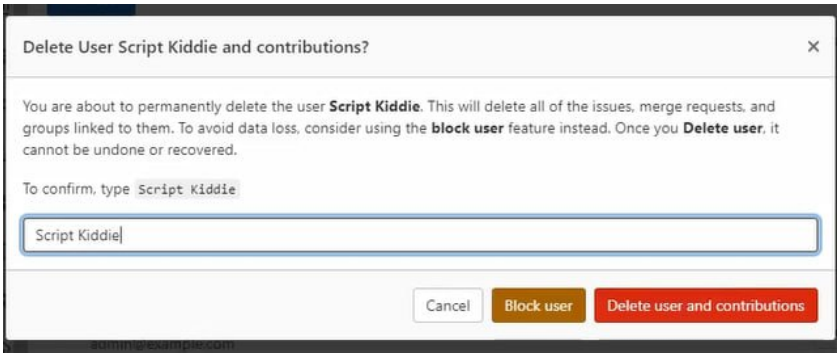

Рис. 3.12: Удаление пользователя Gitlab

Далее переходим к полезной нагрузке, в нашем случае это Meterpreter-сессия, её цель – получение нарушителем Meterpreter-сессии с уязвимым сервером. Для этого нам нужно обнаружить полезную нагрузку:

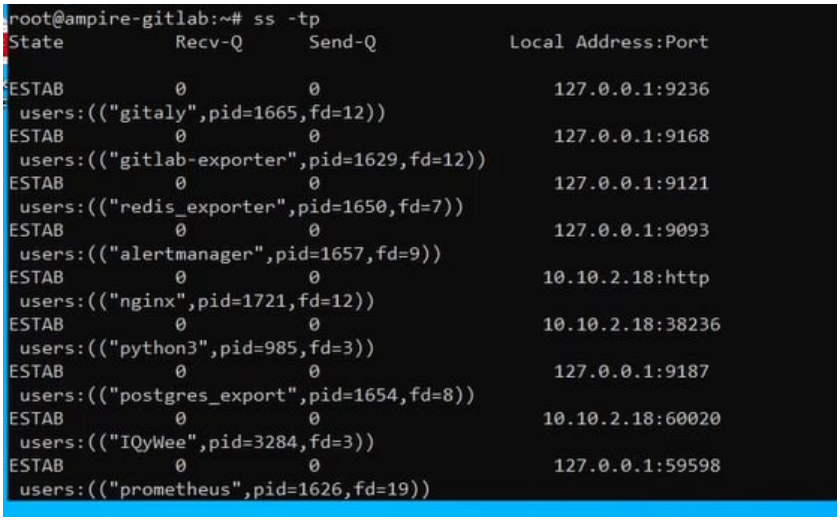

Рис. 3.13: Обнаружение полезной нагрузки

И удалить её:

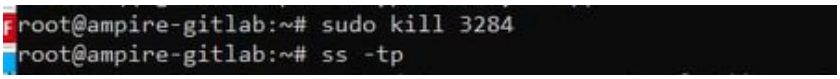

Рис. 3.14: Удаление полезной нагрузки

Теперь закрыта уязвимость:

Рис. 3.15: Закрытие уязвимости

Также закрыто и последствие:

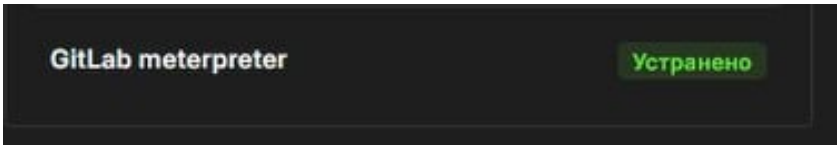

Рис. 3.16: Закрытие последствия

3.3 Уязвимый узел API-Manager

Уязвимость платформы для интеграции интерфейсов прикладного программирования, приложений и веб-служб WSO2 связана с возможностью загрузки произвольного JSP-файла на сервер. Эксплуатация уязвимости может позволить нарушителю, действующему удаленно, выполнить произвольный код.

Сначала заведём карточку с описание уязвимости, ее индикаторами и рекомендациями по устранению.

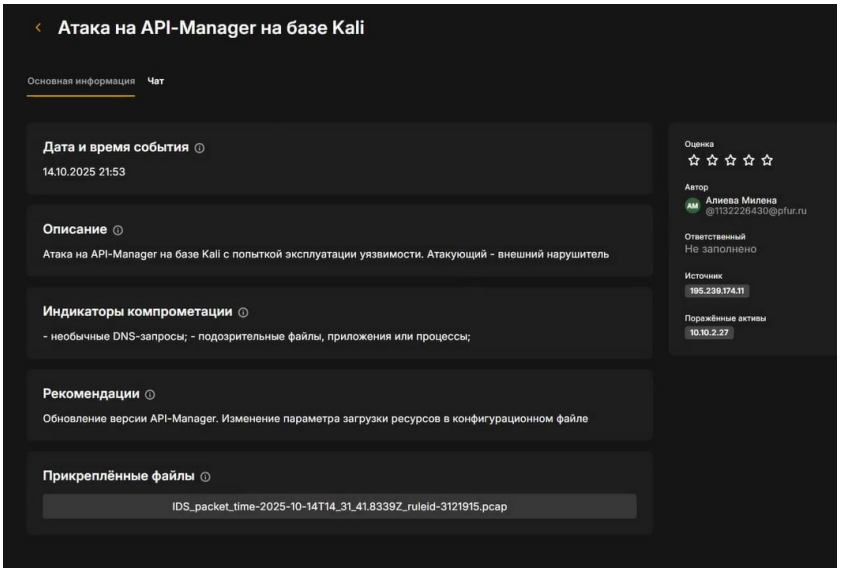

Рис. 3.17: Карточка уязвимости

Далее в нашем случае будем менять параметр загрузки ресурсов в конфигурационном файле. Для этого откроем файл конфигурации WSO2 API-Manager и добавим следующую запись:

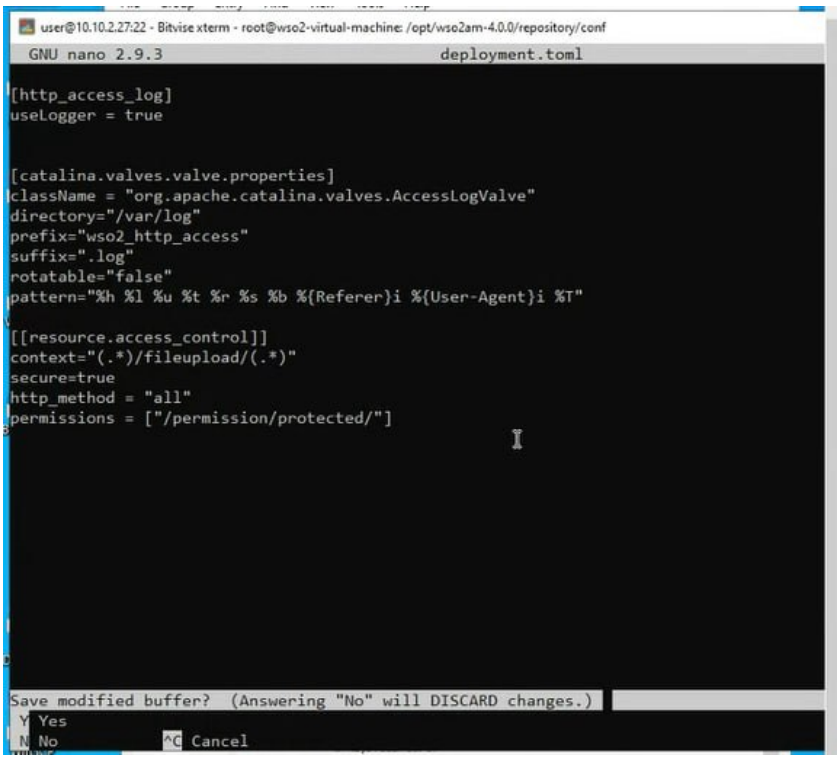

Рис. 3.18: Файл конфигурации WSO2 API-Manager

Для вступления в силу внесенных изменений необходимо перезапустить службу с помощью команды: systemctl restart wso2api.service

Рис. 3.19: Перезапуск службы

Также необходимо удалить загруженный exploit.jsp файл и сгенерированный файл payload.elf, так как наличие данных файлов на атакуемой машине позволит нарушителю получить сессию и после внесения изменений в конфигурационный файл

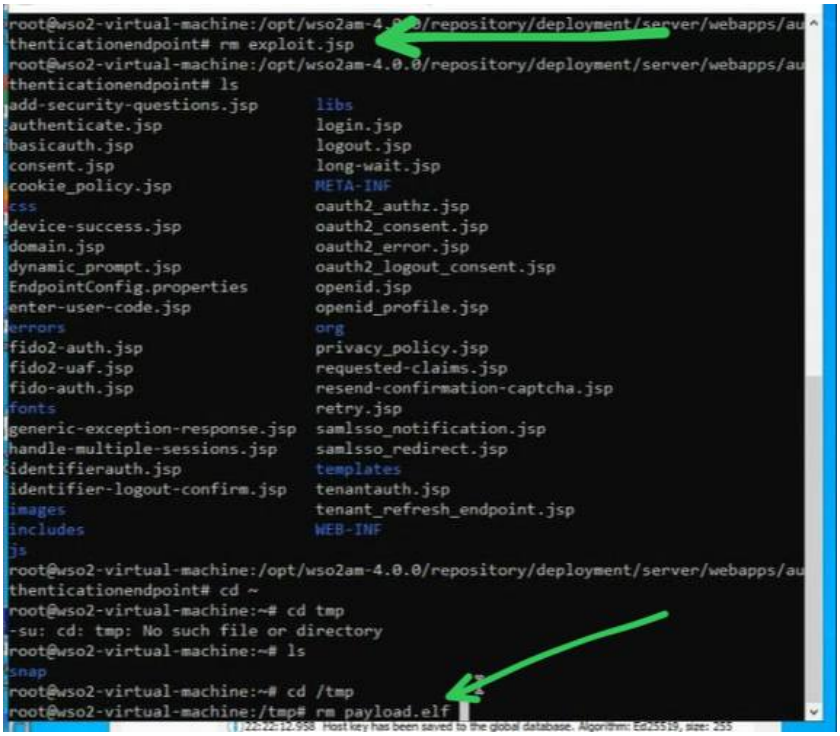

Рис. 3.20: Удаление файлов

Теперь уязвимость устранена:

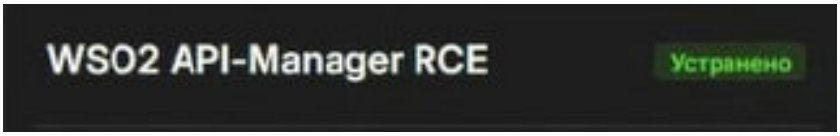

Рис. 3.21: Устранение уязвимости

Далее перейдём к полезной нагрузке. В нашем случае полезная нагрузка заключается в создании нарушителем пользователя в веб-интерфейсе WSO2 API-Manager. Для этого перейдём в веб-интерфейс WSO2 API-Manager

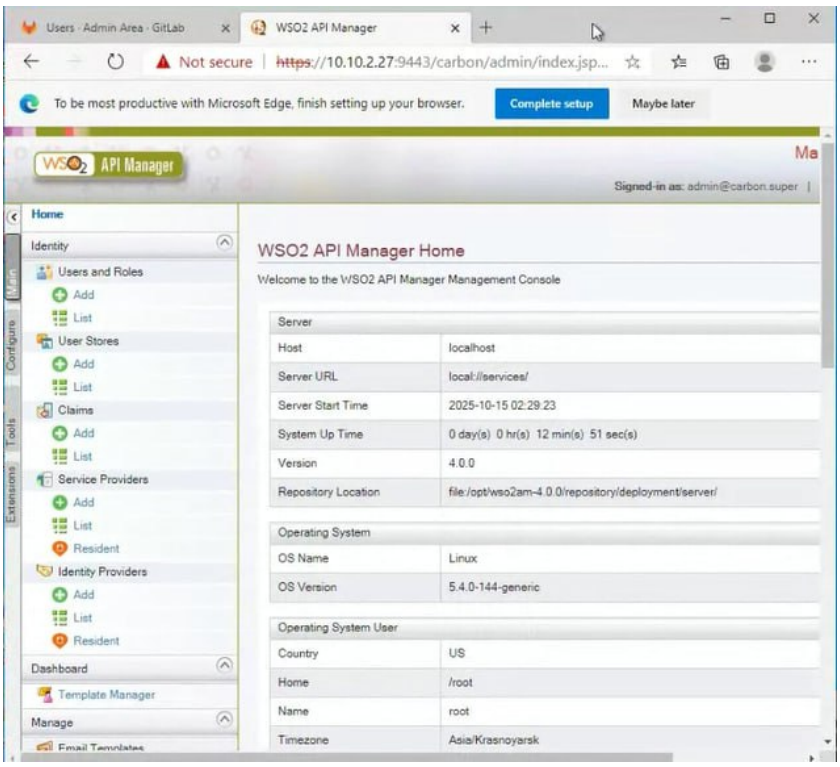

Рис. 3.22: Веб-интерфейс WSO2 API-Manager

Для нейтрализации данной полезной нагрузки необходимо удалить созданного пользователя в веб-интерфейсе

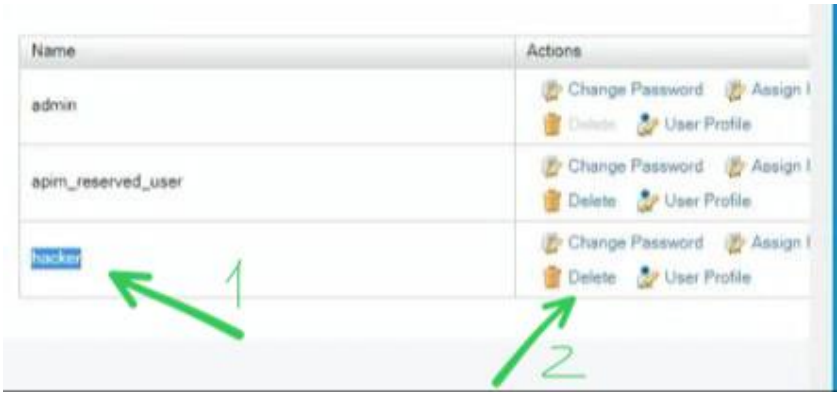

Рис. 3.23: Удаление пользователя

Теперь устранено последствие:

Рис. 3.24: Удаление пользователя

4 Выводы

В ходе выполнения лабораторной работы были выявлены и устранены уязвимости на различные узлы и их последствия, а также система была приведена в безопасное состояние.