

## Выполнение лабораторной работы №2

---

Цель

---

Закрепить практические навыки устранения уязвимостей и защиты интеграционной платформы

## Задание

---

Провести анализ уязвимостей, устранить их и последствия

## 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 описанием уязвимости, ее индикаторами и рекомендациями по устранению.

## Уязвимый узел Bitrix

---



5/37

## Уязвимый узел Bitrix

---

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`

## Уязвимый узел Bitrix

---

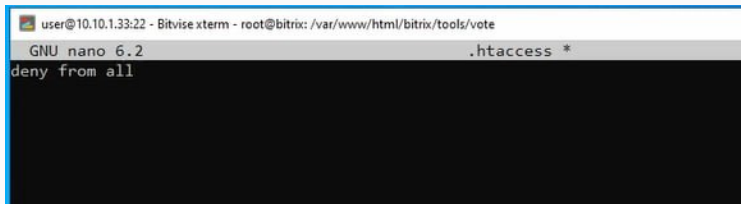
```
root@bitrix:/var/www/html# ls -la
итого 5640
drwxrwxr-x 12 www-data www-data 4096 окт 16 13:10 .
drwxr-xr-x 3 root root 4096 июл 7 2023 ..
-rw-r--r-- 1 www-data www-data 519 июл 7 2023 404.php
-rw-r--r-- 1 www-data www-data 216 июл 7 2023 .access.php
-rwsr-sr-x 1 root root 16048 июл 31 2023 apache_restart
drwxrwxr-x 25 www-data www-data 4096 сен 22 2023 bitrix
-rw-r--r-- 1 www-data www-data 265 июл 7 2023 .bottom.menu.php
-rw-r--r-- 1 www-data www-data 34 окт 16 13:09 caidao.php
drwxr-xr-x 2 www-data www-data 4096 июл 7 2023 company
drwxr-xr-x 2 www-data www-data 4096 июл 7 2023 contacts
-rw-r--r-- 1 www-data www-data 860 июл 7 2023 .htaccess
drwxr-xr-x 2 www-data www-data 4096 июл 7 2023 include
-rw-r--r-- 1 www-data www-data 1168 окт 16 13:10 index.php
drwxr-xr-x 2 www-data www-data 4096 июл 7 2023 login
drwxr-xr-x 2 www-data www-data 4096 июл 7 2023 news
-rw-r--r-- 1 root root 201 окт 16 13:10 password_recovery.php
drwxr-xr-x 2 www-data www-data 4096 июл 7 2023 products
-rw-r--r-- 1 root root 5661008 окт 16 13:10 RickRolled.mp4
-rw-r--r-- 1 www-data www-data 76 окт 16 13:09 script.sh
drwxr-xr-x 2 www-data www-data 4096 июл 7 2023 search
-rw-r--r-- 1 www-data www-data 611 июл 7 2023 .section.php
drwxr-xr-x 2 www-data www-data 4096 июл 7 2023 services
-rw-r--r-- 1 www-data www-data 496 июл 7 2023 .top.menu.php
drwxrwxr-x 4 www-data www-data 4096 окт 16 13:09 upload
-rw-r--r-- 1 www-data www-data 509 июл 7 2023 urlrewrite.php
root@bitrix:/var/www/html# chmod -s apache_restart
```

Рис. 2: Закрывание вектора для локального повышения привелегий

## Уязвимый узел Bitrix

---

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`.



```
user@10.10.1.33:22 - Bitvise xterm - root@bitrix: /var/www/html/bitrix/tools/vote
GNU nano 6.2 .htaccess *
deny from all
```

Рис. 3: Закрытие локального повышения привилегий

## Уязвимый узел Bitrix

---



```
root@bitrix:~# ss -tp
State      Recv-Q    Send-Q      Local Address:Port      Peer Address:Port      Process
ESTAB      0          52          10.10.1.33:ssh          10.10.1.253:65380      users:({("sshd",pid=3627,fd=
ESTAB      0          0          10.10.1.33:40140        195.239.174.11:5557    users:({("apache2",pid=805,fd
CLOSE-WAIT 1          0          [::ffff:10.10.1.33]:http [::ffff:195.239.174.11]:37839 users:({("apache2",pid=805,fd
root@bitrix:~# kill -9 805
```

Рис. 4: Закрытие сессии

## Уязвимый узел Bitrix

---

Теперь уязвимость устранена:

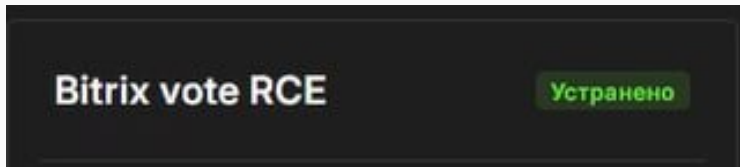
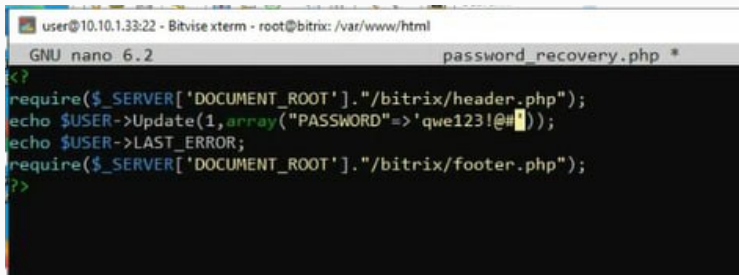


Рис. 5: Заккрытие уязвимости

## Уязвимый узел Bitrix

---

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



```
user@10.10.1.33:22 - Bitvise xterm - root@bitrix: /var/www/html
GNU nano 6.2 password_recovery.php *
<?
require($_SERVER['DOCUMENT_ROOT'].'/bitrix/header.php');
echo $USER->Update(1,array("PASSWORD"=>'qwe123!@#'));
echo $USER->LAST_ERROR;
require($_SERVER['DOCUMENT_ROOT'].'/bitrix/footer.php');
?>
```

Рис. 6: Изменение пароля

## Уязвимый узел Bitrix

---

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.

```
root@bitrix:/var/www/html# rm -r *
root@bitrix:/var/www/html# ls
root@bitrix:/var/www/html# cd ..
root@bitrix:/var/www# cd ..
root@bitrix:/var# cd bitrix_backups/
root@bitrix:/var/bitrix_backups# tar xvfz Bitrix_full_backup.tar.gz -C /var/www/html
```

Рис. 7: Восстановление сайта

## Уязвимый узел Bitrix

---



Теперь последствие устранено:

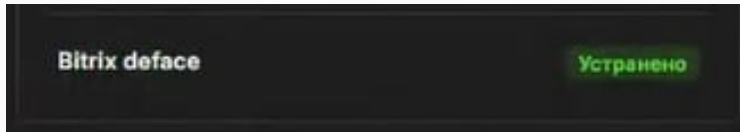


Рис. 8: Заккрытие последствия

## 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.

## Уязвимый узел Gitlab

---

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

## Уязвимый узел Gitlab

---

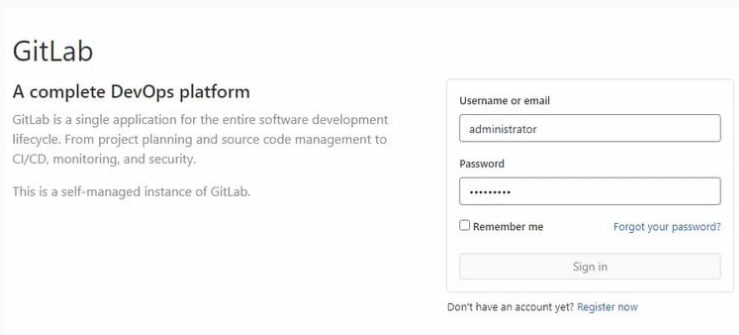
16/37

## Уязвимый узел Gitlab

---



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



The image shows the GitLab login interface. On the left, the GitLab logo is displayed above the text 'A complete DevOps platform'. Below this, a paragraph describes GitLab as a single application for the entire software development lifecycle, from project planning and source code management to CI/CD, monitoring, and security. It also states 'This is a self-managed instance of GitLab.' On the right, there is a login form with two input fields: 'Username or email' containing the text 'administrator' and 'Password' containing a masked password '\*\*\*\*\*'. Below the password field, there is a checkbox for 'Remember me' and a link for 'Forgot your password?'. A 'Sign in' button is located at the bottom of the form. Below the form, there is a link that says 'Don't have an account yet? Register now'.

GitLab

**A complete DevOps platform**

GitLab is a single application for the entire software development lifecycle. From project planning and source code management to CI/CD, monitoring, and security.

This is a self-managed instance of GitLab.

Username or email

administrator

Password

\*\*\*\*\*

☐ Remember me [Forgot your password?](#)

Sign in

Don't have an account yet? [Register now](#)

Рис. 10: Авторизация Gitlab

## Уязвимый узел Gitlab

---

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

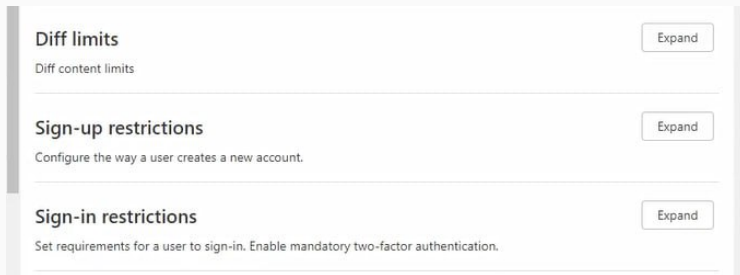


Рис. 11: Настройки Gitlab

## Уязвимый узел 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:

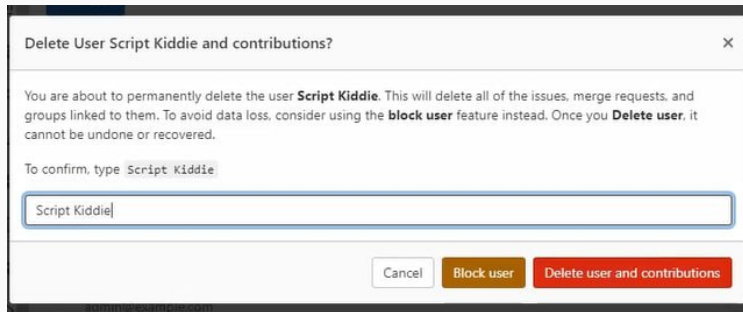


Рис. 12: Удаление пользователя Gitlab

## Уязвимый узел 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:

```
root@ampire-gitlab:~# ss -tp
State      Recv-Q    Send-Q    Local Address:Port
ESTAB      0         0         127.0.0.1:9236
users:((("gitaly",pid=1665,fd=12))
ESTAB      0         0         127.0.0.1:9168
users:((("gitlab-exporter",pid=1629,fd=12))
ESTAB      0         0         127.0.0.1:9121
users:((("redis_exporter",pid=1650,fd=7))
ESTAB      0         0         127.0.0.1:9093
users:((("alertmanager",pid=1657,fd=9))
ESTAB      0         0         10.10.2.18:http
users:((("nginx",pid=1721,fd=12))
ESTAB      0         0         10.10.2.18:38236
users:((("python3",pid=985,fd=3))
ESTAB      0         0         127.0.0.1:9187
users:((("postgres_export",pid=1654,fd=8))
ESTAB      0         0         10.10.2.18:60020
users:((("IQyWee",pid=3284,fd=3))
ESTAB      0         0         127.0.0.1:59598
users:((("prometheus",pid=1626,fd=19))
```

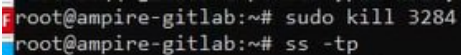
Рис. 13: Обнаружение полезной нагрузки

## Уязвимый узел Gitlab

---



И удалить её:

A terminal window with a black background and white text. The prompt is 'root@ampire-gitlab:~#'. The first command entered is 'sudo kill 3284'. The second command entered is 'ss -tp'.

```
root@ampire-gitlab:~# sudo kill 3284
root@ampire-gitlab:~# ss -tp
```

Рис. 14: Удаление полезной нагрузки

## Уязвимый узел Gitlab

---

Теперь закрыта уязвимость:

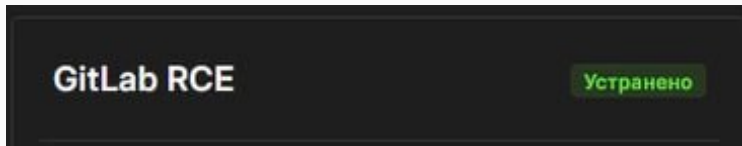


Рис. 15: Заккрытие уязвимости

## Уязвимый узел Gitlab

---

Также закрыто и последствие:

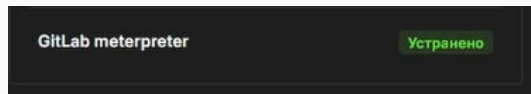


Рис. 16: Закрытие последствия

## 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.

## Уязвимый узел API-Manager

---



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

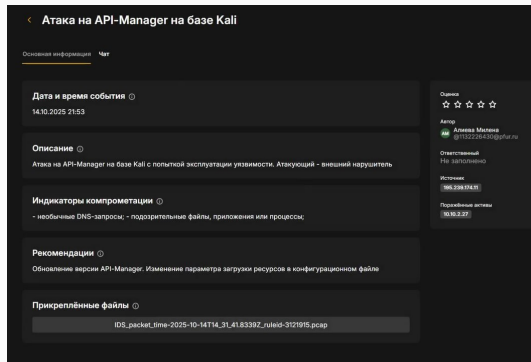


Рис. 17: Карточка уязвимости

## Уязвимый узел API-Manager

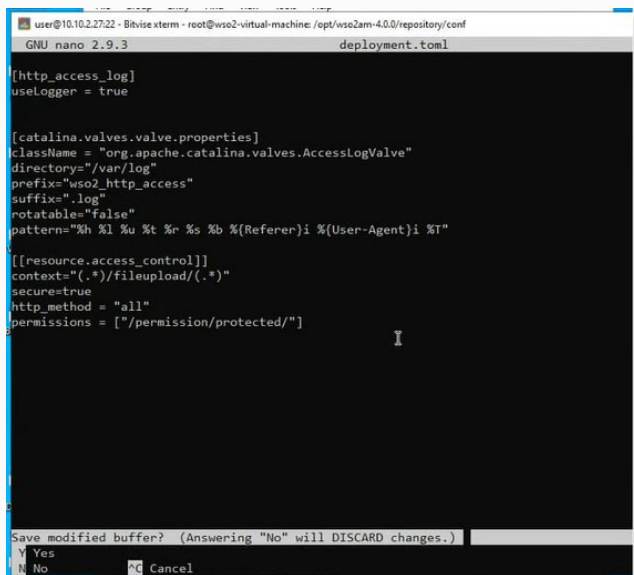
---

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

## Уязвимый узел API-Manager

---

## Уязвимый узел API-Manager



```
user@10.10.2.27:22 - Bitvise xterm - root@wso2-virtual-machine: /opt/wso2am-4.0.0/repository/conf
GNU nano 2.9.3 deployment.toml

[http_access_log]
useLogger = true

[catalina.valves.valve.properties]
className = "org.apache.catalina.valves.AccessLogValve"
directory="/var/log"
prefix="wso2_http_access"
suffix=".log"
rotatable="false"
pattern="%h %l %u %t %r %s %b %{Referer}i %{User-Agent}i %T"

[[resource.access_control]]
context="(.) /fileupload/(.*)"
secure=true
http_method = "all"
permissions = ["/permission/protected/"]

Save modified buffer? (Answering "No" will DISCARD changes.)
Y Yes
N No ^C Cancel
```

## Уязвимый узел API-Manager

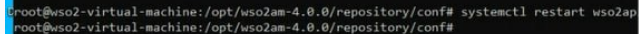
---

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

## Уязвимый узел API-Manager

---





```
root@wso2-virtual-machine:/opt/wso2am-4.0.0/repository/conf# systemctl restart wso2ap
root@wso2-virtual-machine:/opt/wso2am-4.0.0/repository/conf#
```

Рис. 19: Перезапуск службы

## Уязвимый узел API-Manager

---

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

## Уязвимый узел API-Manager

---

## Уязвимый узел API-Manager

```
root@wso2-virtual-machine:/opt/wso2am-4.0.0/repository/deployment/server/webapps/au
thenticationendpoint# rm exploit.jsp
root@wso2-virtual-machine:/opt/wso2am-4.0.0/repository/deployment/server/webapps/au
thenticationendpoint# ls
add-security-questions.jsp      libs
authenticate.jsp                login.jsp
basicauth.jsp                  logout.jsp
consent.jsp                     long-wait.jsp
cookie_policy.jsp              META-INF
css                             oauth2_authz.jsp
device-success.jsp             oauth2_consent.jsp
domain.jsp                     oauth2_error.jsp
dynamic_prompt.jsp             oauth2_logout_consent.jsp
EndpointConfig.properties      openid.jsp
enter-user-code.jsp            openid_profile.jsp
errors                          org
fido2-auth.jsp                 privacy_policy.jsp
fido2-uaf.jsp                  requested-claims.jsp
fido-auth.jsp                  resend_confirmation_captcha.jsp
fonts                           retry.jsp
generic-exception-response.jsp samlso_notification.jsp
handle-multiple-sessions.jsp   samlso_redirect.jsp
identifierauth.jsp             templates
identifier-logout-confirm.jsp  tenantauth.jsp
images                         tenant_refresh_endpoint.jsp
includes                       WEB-INF
js
root@wso2-virtual-machine:/opt/wso2am-4.0.0/repository/deployment/server/webapps/au
thenticationendpoint# cd ~
root@wso2-virtual-machine:~# cd tmp
-su: cd: tmp: No such file or directory
root@wso2-virtual-machine:~# ls
snap
root@wso2-virtual-machine:~# cd /tmp
root@wso2-virtual-machine:/tmp# rm payload.elf
```

Рис. 20: Удаление файлов

## Уязвимый узел API-Manager

---

Теперь уязвимость устранена:

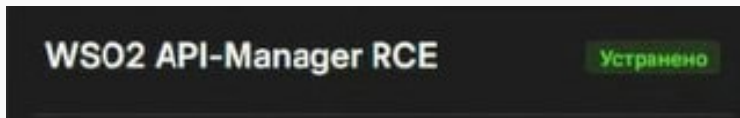


Рис. 21: Устранение уязвимости

## Уязвимый узел API-Manager

---



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

## Уязвимый узел API-Manager

---

# Уязвимый узел API-Manager

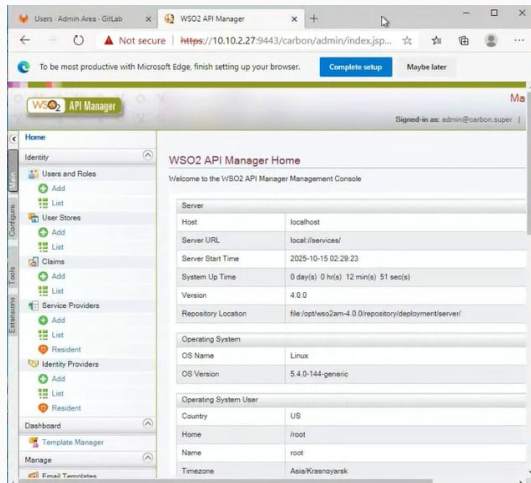


Рис. 22: Веб-интерфейс WSO2 API-Manager

## Уязвимый узел API-Manager

---

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

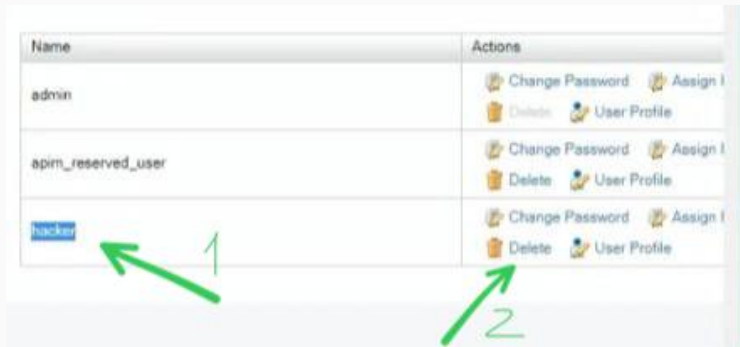


Рис. 23: Удаление пользователя

## Уязвимый узел API-Manager

---

Теперь устранено последствие:

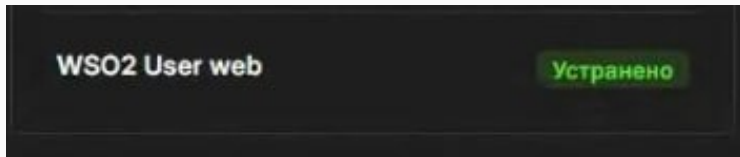


Рис. 24: Удаление пользователя

## 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