



Программный комплекс кибертренировок «Солар Кибермир»

Инструкция по установке и настройке

МОСКВА, 2023

Содержание

1.	ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ.....	4
1.1.	НАЗНАЧЕНИЕ	4
1.2.	УСЛОВИЯ ЭКСПЛУАТАЦИИ.....	4
1.2.1.	ТРЕБОВАНИЯ К СЕРВЕРУ ПРИЛОЖЕНИЯ.....	4
1.2.2.	ТРЕБОВАНИЯ К АВТОМАТИЗИРОВАННОМУ РАБОЧЕМУ МЕСТУ ПОЛЬЗОВАТЕЛЯ.....	4
2.	ПОРЯДОК НАСТРОЙКИ.....	6
2.1.	ОБЩИЕ СВЕДЕНИЯ	6
2.2.	УСТАНОВКА JAVA 11.....	6
2.2.1.	ОБНОВЛЕНИЕ ПАКЕТОВ	6
2.2.2.	УСТАНОВКА JAVA 11.....	6
2.3.	УСТАНОВКА DOCKER/DOCKER-COMPOSE	7
2.3.1.	УСТАНОВКА DOCKER.....	7
2.3.2.	УСТАНОВКА DOCKER-COMPOSE.....	7
2.4.	СОЗДАНИЕ УЧЕТНОЙ ЗАПИСИ ПОЛЬЗОВАТЕЛЯ И ГРУППЫ CYBERRANGE	7
2.4.1.	СОЗДАНИЕ УЧЕТНОЙ ЗАПИСИ CYBERRANGE	7
2.4.2.	СОЗДАНИЕ ДИРЕКТОРИИ «NG»	7
2.5.	ПРОЦЕСС РАЗВЕРТЫВАНИЯ CYBERRANGE.....	7
2.5.1.	КОПИРОВАНИЕ ФАЙЛОВ И ДИРЕКТОРИЙ В /HOME/CYBERRANGE/NG/	7
2.5.2.	НАЗНАЧЕНИЕ ПОЛЬЗОВАТЕЛЮ ПРАВ OWNER.....	8
2.5.3.	НАСТРОЙКА КОНФИГУРАЦИОННЫХ ФАЙЛОВ.....	8
2.5.4.	КОПИРОВАНИЕ СЕРВИСОВ В ETC/SYSTEMD/SYSTEM.....	9
2.5.5.	ВКЛЮЧЕНИЕ SYSTEMD-СЕРВИСОВ.....	10
2.5.6.	ИЗМЕНЕНИЕ ПРАВ ДЛЯ *.SH-ФАЙЛОВ.....	10
2.6.	ПРОЦЕСС РАЗВЕРТЫВАНИЯ CONTENT SERVICE.....	10
2.6.1.	КОПИРОВАНИЕ ФАЙЛОВ И ДИРЕКТОРИЙ В /OPT/CONTENTS/	10
2.6.2.	НАСТРОЙКА КОНФИГУРАЦИОННЫХ ФАЙЛОВ.....	11
2.6.3.	КОПИРОВАНИЕ ФАЙЛА CONTENT-SERVICE-V3.3.SERVICE В /ETC/SYSTEMD/SYSTEM/12	
2.7.	ПРОЦЕСС РАЗВЕРТЫВАНИЕ GUACAMOLE PROXY.....	13
2.7.1.	КОПИРОВАНИЕ ФАЙЛОВ И ДИРЕКТОРИЙ В /OPT/CONTENTS/	13
2.7.2.	НАСТРОЙКА КОНФИГУРАЦИОННЫХ ФАЙЛОВ.....	14

2.7.3.	УСТАНОВКА REDIS.....	15
2.7.4.	КОПИРОВАНИЕ ФАЙЛА GUACAMOLE-PROXY.SERVICE В /ETC/SYSTEMD/SYSTEM/	15
2.8.	ЗАПУСК СЕРВИСОВ.....	16
2.8.1.	ЗАПУСК GUACAMOLE-PROXY.SERVICE.....	16
2.8.2.	ЗАПУСК CONTENT-SERVICE-V3.3.SERVICE	16
2.8.3.	ЗАПУСК CYBERRANGE-IFX.SERVICE.....	17
2.8.4.	ЗАПУСК CYBERRANGE-NG.SERVICE	17

1. Общие сведения о программе

1.1. Назначение

Программный комплекс кибертренировок «Солар Кибермир» предназначен для повышения уровня практической подготовки специалистов в области информационной безопасности по выявлению компьютерных атак, расследованию инцидентов информационной безопасности за счет автоматизации следующих процессов:

- проведение практических занятий по информационной безопасности (далее – ИБ): обнаружение, расследование и защита от кибератак (далее – КА);
- проведение учений и соревнований по информационной безопасности и защите от КА для специалистов по ИБ;
- оценка практических навыков специалистов в области ИБ;
- проведение исследований в области информационной безопасности программного обеспечения и автоматизированных систем (далее – ИС).

1.2. Условия эксплуатации

1.2.1. Требования к серверу приложения

Минимальные требования к аппаратному обеспечению сервера приложений:

- процессор не менее 12 ядер с тактовой частотой не менее 3 ГГц и поддержкой архитектуры x64;
- объем оперативной памяти не менее 32 ГБ;
- объем жесткого диска не менее 250 ГБ;
- сетевой адаптер Ethernet с пропускной способностью не ниже 100 Мбит/с.

Требования к программному обеспечению сервера:

- операционная система Linux x64 без графического интерфейса с установленными средствами защиты информации.

1.2.2. Требования к автоматизированному рабочему месту пользователя

Требования к аппаратному обеспечению:

- процессор не менее 4 ядер с тактовой частотой не менее 2,8 ГГц;
- объем оперативной памяти не менее 4 ГБ;
- объем жесткого диска не менее 128 ГБ;

- разрешение экрана при работе с интерфейсом не менее 1024x768 (интерфейс оптимизирован для разрешения 1920x1080);
- сетевой адаптер Ethernet с пропускной способностью не ниже 10 Мбит/с.

Требования к программному обеспечению:

- операционная система Linux x64 с установленным графическим интерфейсом.

2. Порядок настройки

2.1. Общие сведения

Для выполнения настройки необходимо следующее ПО и ОС:

- Java 11;
- Docker;
- Docker-compose;
- Redis;
- ОС Linux.

2.2. Установка Java 11

2.2.1. Обновление пакетов

Для установки JAVA 11 необходимо обновить указатель пакетов.

Выполните команды:

```
sudo apt update;  
sudo apt upgrade.
```

2.2.2. Установка JAVA 11

1. Проверьте, установлена ли Java на сервере. Для этого выполните команду:
`java-version`.
2. Для установки выполните команду:
`sudo apt install openjdk-11-jre-headless`.
3. Для проверки установки выполните команду:
`java-version`.

Пример установки JAVA 11 представлен на Рис. 1:

```
openjdk version "11.0.17" 2022-10-18  
OpenJDK Runtime Environment (build 11.0.17+8-post-Ubuntu-lubuntu218.04)  
OpenJDK 64-Bit Server VM (build 11.0.17+8-post-Ubuntu-lubuntu218.04, mixed mode, sharing)  
[demo3:~$
```

Рис. 1 – Пример установки JAVA

Ссылка на официальный ресурс <https://docs.oracle.com/javase/10/install/installation-jdk-and-jre-linux-platforms.htm#JSJIG-GUID-737A84E4-2EFF-4D38-8E60-3E29D1B884B8>

2.3. Установка Docker/Docker-compose

2.3.1. Установка Docker

Установите Docker, как указано в инструкции:

<https://docs.docker.com/engine/install/ubuntu>

2.3.2. Установка Docker-compose

Установите Docker-compose, как указано в инструкции:

<https://docs.docker.com/compose/install/linux/>

2.4. Создание учетной записи пользователя и группы cyberrange

2.4.1. Создание учетной записи cyberrange

1. Для создания учетной записи пользователя выполните команду:
`sudo adduser cyberrange.`
2. Для добавления пользователя в группу администраторов выполните команду:
`sudo usermod -aG sudo cyberrange.`
3. Добавьте пользователя в докер, для чего выполните команду:
`sudo adduser cyberrange docker.`

2.4.2. Создание директории «ng»

1. Для создания директории перейдите в директорию:
`cd/home/cyberrrange.`
2. Для создания директории ng выполните команду
`mkdir ng.`

2.5. Процесс развертывания cyberrange

2.5.1. Копирование файлов и директорий в /home/cyberrange/ng/

1. Перейдите в директорию /home/cyberrange/ng/ .
2. В архиве дистрибутива выберите следующие файлы и директории:
 - директория config;
 - директория docker4;
 - директория etc;
 - директория files;

- директория lib;
 - директория static;
 - Файл cyberrange;
 - Файл cyberrange.env.
3. Скопируйте выбранные файлы и директории в /home/cyberrange/ng/.

2.5.2. Назначение пользователю прав owner

1. Для назначения пользователю cyberrange прав owner выполните команду:
`sudo chown -R cyberrange:cyberrange ~cyberrange/ng.`
2. Перейдите в каталог ~cyberrange/ng/.
3. Сделайте файл cyberrange исполняемым. Для этого выполните команду:
`sudo chmod +x cyberrange.`

2.5.3. Настройка конфигурационных файлов

1. Для настройки конфигурационных файлов отредактируйте следующие файлы:
`cyberrange.env;`
`cyberrange-ifx.service.template;`
`cyberrange-ng.service.template.`
2. В файле `cyberrange.env` отредактируйте параметры:
`export CR_BASE_URL_HOST= хост (см. Рис. 2)`

```
# per-host parameters
export CR_BASE_URL_PROTO=https
export CR_BASE_URL_HOST=demo4.cyber.rt-solar.ru
export CR_BASE_URL_PORT=443
#export CR_BASE_URL=${CR_BASE_URL_PROTO}://${CR_BASE_URL_HOST}:${CR_BASE_URL_PORT}
```

Рис. 2 – Редактирование CR_BASE_URL_HOST

3. `export CR_VERSION= версия (см. Рис. 3)`

```
# system version
export CR_VERSION=3.3.0
```

Рис. 3 – Редактирование CR_VERSION

4. В файле `cyberrange-ifx.service.template` отредактируйте параметры, как показано на Рис. 4:
`ExecStart=/home/cyberrange/ng/docker/start.sh;`
`ExecStop=/home/cyberrange/ng/docker/stop.sh;`

WorkingDirectory=/home/cyberrange/ng/docker.

```
[Unit]
Description=Cyberrange (NG) infrastructure services
After=syslog.target network.target postgresql.target

[Service]
TasksMax=65535
Type=forking
SuccessExitStatus=143
SuccessExitStatus=1
ExecStart=/home/cyberrange/ng/docker/start.sh
ExecStop=/home/cyberrange/ng/docker/stop.sh
User=cyberrange
Group=cyberrange
WorkingDirectory=/home/cyberrange/ng/docker

[Install]
WantedBy=multi-user.target
```

Рис. 4 – Редактирование файла cyberrange-ifx.service.template

5. В файле cyberrange-ng.service.template отредактируйте параметры, как показано на Рис. 5:

ExecStart=/home/cyberrange/ng/cyberrange start;

ExecStop=/home/cyberrange/ng/cyberrange stop;

WorkingDirectory=/home/cyberrange/ng.

```
[Unit]
Description=Cyberrange (NG) application
After=syslog.target network.target postgresql.target cyberrange-ifx.service

[Service]
Type=forking
SuccessExitStatus=143
ExecStart=/home/cyberrange/ng/cyberrange start
ExecStop=/home/cyberrange/ng/cyberrange stop
User=cyberrange
Group=cyberrange
WorkingDirectory=/home/cyberrange/ng

[Install]
WantedBy=multi-user.target
```

Рис. 5 – Редактирование файла cyberrange-ng.service.template

2.5.4. Копирование сервисов в etc/systemd/system

1. Откройте проводник.
2. Перейдите в директорию etc/systemd/system.

3. Переименуйте файлы:

```
cyberrange-ng.service.template -> cyberrange-ng.service;
cyberrange-ifx.service.template -> cyberrange-ifx.service.
```

4. Скопируйте файлы `cyberrange-ng.service` и `cyberrange-ifx.service` в `etc/systemd/system`, как показано на **Рис. 6**

/etc/systemd/system				/home/cyberrange/ng/etc/systemd/system			
Name	Size	Modify time		Name	Size	Modify time	
UP--DIR		Nov 7 12:56		UP--DIR		Nov 10 11:50	
cloud-final.service.wants	4096	Sep 15 2021		cyberrange-ifx.service.template	398	Nov 10 11:58	
cloud-init.target.wants	4096	Nov 7 12:50		cyberrange-ng.service.template	369	Oct 7 14:58	
default.target.wants	4096	Nov 9 14:41					
final.target.wants	4096	Sep 15 2021					
getty.target.wants	4096	Sep 15 2021					
graphical.target.wants	4096	Sep 15 2021					
multi-user.target.wants	4096	Nov 9 14:54					
network-online.target.wants	4096	Sep 15 2021					
open-vm-tools.service.requires	4096	Sep 15 2021					
paths.target.wants	4096	Sep 15 2021					
sockets.target.wants	4096	Nov 9 14:54					
ssh-keygen@.service.d	4096	Nov 7 13:08					
sysinit.target.wants	4096	Sep 15 2021					
timers.target.wants	4096	Nov 7 13:08					
cyberrange-ifx.service	398	Nov 10 11:58					
cyberrange-ng.service	369	Oct 7 14:50					
@dbus-org.freedesktop.resolve1.service	44	Sep 15 2021					
@dbus-org.freedesktop.themald.service	36	Nov 7 12:46					
@iscsi.service	38	Sep 15 2021					
snap-core18-2620.mount	300	Nov 9 14:41					
snap-docker-2285.mount	300	Nov 9 14:41					
snap-snapd-17336.mount	300	Nov 9 14:41					
snap.docker.dockerd.service	486	Nov 9 14:41					
sshd.service	31	Nov 7 12:47					
syslog.service	35	Sep 15 2021					
vmtoolsd.service	41	Sep 15 2021					

Рис. 6 – Копирование сервисов в `etc/systemd/system`2.5.5. Включение `systemd`-сервисов

Для добавления в автозагрузку сервисов выполните следующие команды:

```
sudo systemctl daemon-reload;
sudo systemctl enable cyberrange-ng.service;
sudo systemctl enable cyberrange-ifx.service.
```

2.5.6. Изменение прав для `*.sh`-файлов

Чтобы сделать `*.sh`-файлы исполняемыми, выполните следующие шаги:

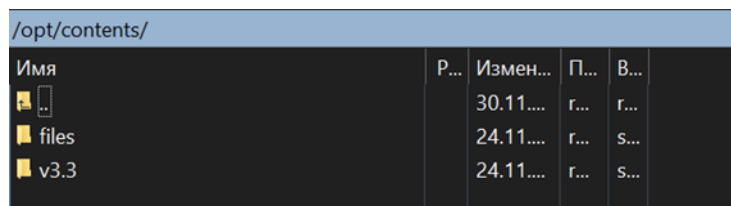
1. Перейдите в директорию `cd/home/cyberrange/ng/docker/`.
2. Нажмите Enter и перейдите в `/home/cyberrange/ng/docker`.
3. Выполните команду:

```
sudo chmod +x *.sh.
```

2.6. Процесс развертывания `content service`2.6.1. Копирование файлов и директорий в `/opt/contents/`

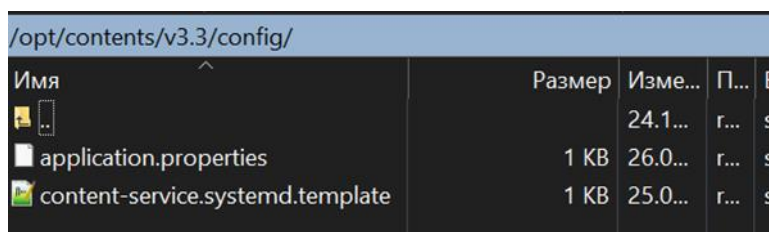
1. Перейдите в `/opt/contents/`.

2. Скопируйте из архива дистрибутива директории `files` и `v3.3` в `/opt/contents/`, как показано на Рис.7

Рис. 7 – Копирование файлов в `/opt/contents/`

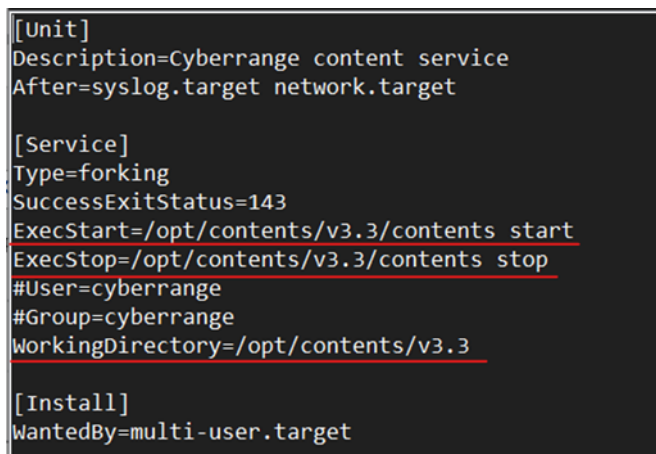
2.6.2. Настройка конфигурационных файлов

1. Перейдите в `/opt/contents/v3.3/config/`, как показано на Рис. 8

Рис. 8 – Переход в `/opt/contents/v3.3/config/`

2. Отредактируйте файл `content-service.systemd.template`, как представлено на Рис. 9:

```
ExecStart=/opt/contents/v3.3/contents start;
ExecStop=/opt/contents/v3.3/contents stop;
WorkingDirectory=/opt/contents/v3.3.
```

Рис. 9 – Редактирование файла `content-service.systemd.template`

3. Настройте внутренний `content service`, как показано на Рис. 10.
4. Выполните команду:


```
cd/home/cyberrange/ng/config/config/cr-config/app/content.
```

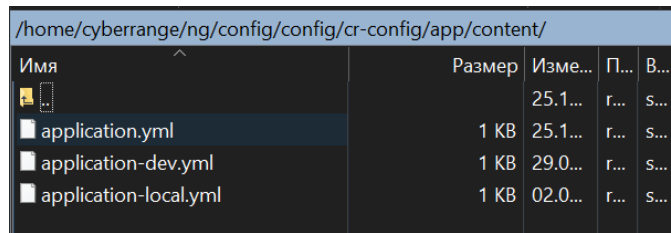


Рис. 10 – Редактирование файла content-service.systemd.template

5. Отредактируйте файл application.yml.

В файле необходимо указать username, password, url, как показано на Рис. 11. Данные можно указать самостоятельно или взять из конфигурационного файла /opt/contents/v3.3/contents.env.

```
cyberange:
  contents:
    username: "test_user"
    password: "test_pass"
    url: "http://localhost:8082"
```

Рис. 11 – Редактирование файла application.yml

6. Сделайте файл /opt/contents/v3.3/contents исполняемым. Для этого:

- а) перейдите в cd/opt/contents/v3.3/;
- б) выполните команду:

```
sudo chmod +x contents.
```

2.6.3. Копирование файла content-service-v3.3.service в /etc/systemd/system/

1. Перейдите в директорию /etc/systemd/system/.
2. Переименуйте файл content-service.systemd.template в content-service-v3.3.service.
3. Скопируйте в etc/systemd/system, как показано на Рис. 12.

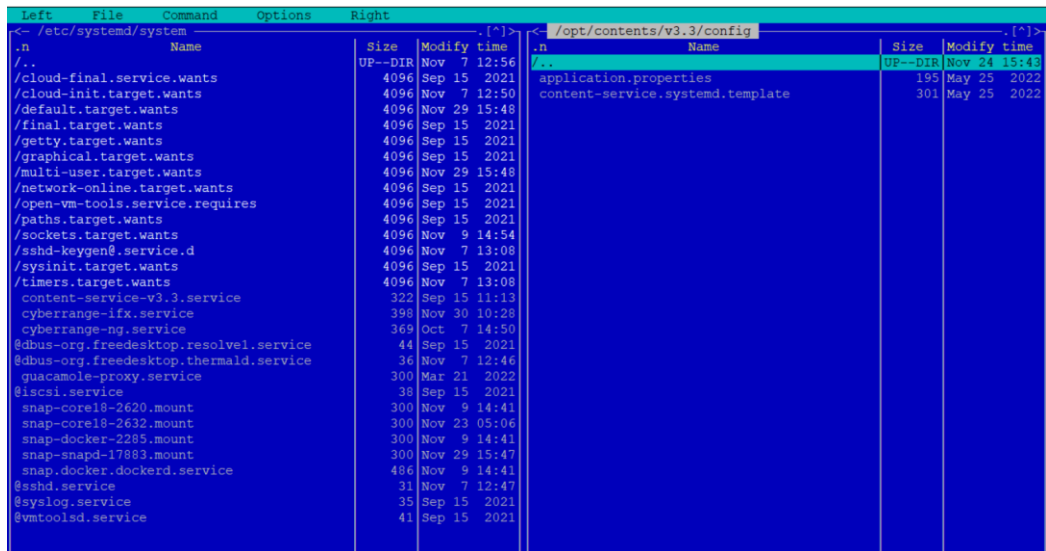


Рис. 12 – Копирование файла content-service.systemd.template

4. Добавьте content-service-v3.3.service в загрузку.

Для этого выполните команду:

```
sudo systemctl enable content-service-v3.3.service.
```

2.7. Процесс развертывание guacamole proxy

2.7.1. Копирование файлов и директорий в /opt/contents/

!!! Рекомендуем устанавливать Guacamole proxy и guacamole daemon на отдельном сервере.

1. Guacamole proxy необходимо развернуть в той же директории, где развернут guacamole daemon.
2. Скопируйте файлы и директории из архива с дистрибутивом в /opt/guacp/, как показано на Рис. 13.

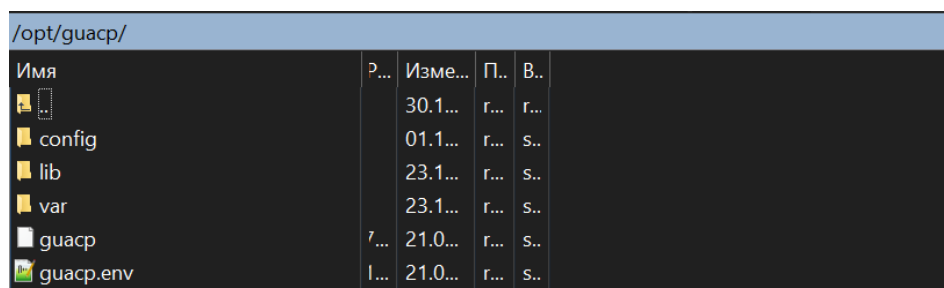


Рис. 13– Копирование файлов в директорию /opt/guacp/

2.7.2. Настройка конфигурационных файлов

1. Перейдите в `/opt/guacp/config/`.
2. Выполните команду:
`cd/opt/guacp/config/`.
3. Отредактируйте файл `guacamole-proxy.systemd.template`, как показано на Рис. 14.
 - а) `ExecStart=/opt/guacp/guacp start`;
 - б) `ExecStop=/opt/guacp/guacp stop`;
 - в) `WorkingDirectory=/opt/guacp`.

```
[Unit]
Description=Cyberange guacamole proxy service
After=syslog.target network.target

[Service]
Type=forking
SuccessExitStatus=143
ExecStart=/opt/guacp/guacp start
ExecStop=/opt/guacp/guacp stop
#User=cyberange
#Group=cyberange
WorkingDirectory=/opt/guacp

[Install]
WantedBy=multi-user.target
```

Рис. 14– Редактирование файла `guacamole-proxy.systemd.template`

4. Перейдите в `cd/home/cyberange/ng/`.
5. Отредактируйте файл `cyberange.env`.
В файле для `guacamole proxy` необходимо указать `username`, `password`, `url`, как показано на Рис. 15
Данные указаны в файле `guacp.env (/opt/guacp/)`.

```
export CR_GUAC_PROXY_URL=http://localhost:8081
export CR_GUAC_PROXY_LOGIN=test_user
export CR_GUAC_PROXY_PASSWORD=test_pass
```

Рис. 15 – Редактирование файла `cyberange.env`

6. Настройте event сервис. Для этого:
 - а) перейдите в директорию
`cd/home/cyberange/ng/config/config/cr-config/app/event/;`
 - б) откройте файл `application.yml` и проверьте данные для подключения к `guacamole proxy`, как показано на Рис. 16.

```

guacamole:
  base-uri: ${CR_GUAC_PROXY_URL:http://192.168.100.201:8081}
  authn:
    basic:
      username: ${CR_GUAC_PROXY_LOGIN:test_user}
      password: ${CR_GUAC_PROXY_PASSWORD:test_pass}

```

Рис. 16 – Проверка файла application.yml

в) сделайте файл `/opt/guacp/guacp` исполняемым. Для этого перейдите в `cd/opt/guacp/` и выполните команду:

```
sudo chmod +x guacp;
```

г) перейдите в `/opt/guacp/config/`: `cd/opt/guacp/config/`;

д) проверьте данные (хост и порт) для доступа к `guacamole daemon`. Для этого откройте файл `application.properties`, как показано на Рис. 17.

```

/opt/guacp/config/application.properties [----] 0 L:[ 1+16 17/ 17] *(56
# guacamole proxy service port
server.port=8081
server.servlet.context-path=/api

# guacamole daemon connection properties
cyberrange.guacamole.host=${GP_HOST:127.0.0.1}
cyberrange.guacamole.port=${GP_PORT:4822}

# guacamole proxy credentials
cyberrange.guacamole.authn.basic.username=${GP_USER:user}
cyberrange.guacamole.authn.basic.password=${GP_PASS:changeit}

# guacamole proxy additional configuration parameters
#cyberrange.guacamole.tunnel.storage.ttl=${GP_TUNNEL_TTL:PT1M}

# redis configuration ?

```

Рис. 17– Проверка данных для доступа к guacamole daemon

2.7.3. Установка Redis

Установите Redis, как указано в инструкции:

<https://redis.io/docs/getting-started/installation/install-redis-on-linux/>

2.7.4. Копирование файла guacamole-proxy.service в /etc/systemd/system/

1. Откройте проводник.
2. Перейдите в директорию `/opt/guacp/config/`.
3. Переименуйте файл `guacamole-proxy.systemd.template` в `guacamole-proxy.service`.
4. Скопируйте файл в `etc/systemd/system`, как показано на Рис. 18.

```

Left File Command Options Right
< /etc/systemd/system .{ } >
Name Size Modify time Name Size Modify time
./ UP--DIR Nov 7 12:58 ./ UP--DIR Nov 30 10:02
/cloud-final.service.wants 4096 Sep 15 2021 application-dev.properties
/cloud-init.target.wants 4096 Nov 7 12:50 application.properties
/default.target.wants 4096 Nov 29 15:48 application.yml
/final.target.wants 4096 Sep 15 2021 guacamole-proxy.service.template
/getty.target.wants 4096 Sep 15 2021
/graphical.target.wants 4096 Sep 15 2021
/multi-user.target.wants 4096 Nov 29 15:48
/network-online.target.wants 4096 Sep 15 2021
/open-vm-tools.service.requires 4096 Sep 15 2021
/paths.target.wants 4096 Sep 15 2021
/sockets.target.wants 4096 Nov 9 14:54
/ssh-keygen.service.d 4096 Nov 7 13:08
/sysinit.target.wants 4096 Sep 15 2021
/timers.target.wants 4096 Nov 7 13:08
content-service-v3.3.service 322 Sep 15 11:13
cyberrange-ifs.service 398 Nov 30 10:28
cyberrange-ng.service 369 Oct 7 14:50
@dbus-org.freedesktop.resolve1.service 44 Sep 15 2021
@dbus-org.freedesktop.thermal.service 38 Nov 7 12:46
guacamole-proxy.service 300 Mar 21 2022
iscsi.service 38 Sep 15 2021
snap-core18-2620.mount 300 Nov 9 14:41
snap-core18-2632.mount 300 Nov 23 05:06
snap-docker-2285.mount 300 Nov 9 14:41
snap-snapd-17883.mount 300 Nov 29 15:47
snap-docker-dockerd.service 488 Nov 9 14:41
@sshd.service 31 Nov 7 12:47
@syslog.service 35 Sep 15 2021
@vmttoolsd.service 41 Sep 15 2021

```

Рис. 18 – Копирование файла guacamole-proxy.service.template

5. Добавьте файл guacamole-proxy.service в загрузку. Для этого выполните команду:

```
sudo systemctl enable guacamole-proxy.service.
```

2.8. Запуск сервисов

2.8.1. Запуск guacamole-proxy.service

Для запуска guacamole-proxy.service нужно выполнить следующие команды:

- sudo systemctl start guacamole-proxy.service;
- sudo systemctl status guacamole-proxy.service.

6. Запуск guacamole-proxy.service представлен на

Рис. 19.

```

• guacamole-proxy.service - Cyberrange guacamole proxy service
  Loaded: loaded (/etc/systemd/system/guacamole-proxy.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2022-12-01 10:35:42 UTC; 6 days ago
    Main PID: 6313 (java)
      Tasks: 68 (limit: 4915)
     CGroup: /system.slice/guacamole-proxy.service
             └─6313 /usr/bin/java -Xmx1g -Xms128m -XX:HeapDumpPath=dumps -XX:+HeapDumpOnOutOfMemoryError -jar -Dspring.profiles.
Dec 01 10:35:41 demo3 systemd[1]: Starting Cyberrange guacamole proxy service...
Dec 01 10:35:41 demo3 guacp[6259]: starting guacamole-proxy
Dec 01 10:35:42 demo3 guacp[6259]: guacamole-proxy service started (PID: 6313)
Dec 01 10:35:42 demo3 systemd[1]: Started Cyberrange guacamole proxy service.
lines 1-12/12 (END)

```

Рис. 19– Процесс запуска guacamole-proxy.service

2.8.2. Запуск content-service-v3.3.service

Для запуска content-service-v3.3.service нужно выполнить следующие команды:

- sudo systemctl start content-service-v3.3.service;
- sudo systemctl status content-service-v3.3.service.

Запуск content-service-v3.3.service представлен на Рис. 20.

```

* content-service-v3.3.service - Cyberrange content service
   loaded: loaded (/etc/systemd/system/content-service-v3.3.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-12-01 10:38:51 UTC; 6 days ago
   Main PID: 7096 (java)
   Tasks: 35 (limit: 4915)
   CGroup: /system.slice/content-service-v3.3.service
           └─7096 /usr/bin/java -Xmx1g -Xms128m -XX:HeapDumpPath=dumps -XX:+HeapDumpOnOutOfMemoryError -jar -Dspring.profiles.
Dec 01 10:38:50 demo3 systemd[1]: Starting Cyberrange content service..
Dec 01 10:38:50 demo3 contents[7063]: starting content-service-v3.3
Dec 01 10:38:51 demo3 contents[7063]: content-service-v3.3 service started (PID: )
Dec 01 10:38:51 demo3 systemd[1]: Started Cyberrange content service.
lines 1-12/12 (END)

```

Рис. 20– Процесс запуска content-service-v3.3.service

2.8.3. Запуск cyberrange-ifx.service

Для запуска cyberrange-ifx.service (сервисы инфраструктуры) нужно выполнить следующие команды:

- а) `sudo adduser cyberrange docker;`
- б) `sudo systemctl start cyberrange-ifx.service.`

Проверьте `docker ps`. Для этого выполните команду `sudo systemctl status cyberrange-ifx.service`, как показано на Рис. 21.

```

CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                                     NAMES
e336d582c393   postgres:10   "docker-entrypoint.s..." 3 hours ago   Up 3 hours   0.0.0.0:5432->5432/tcp, :::5432->5432/tcp   cr_postgres
1104b1ffe52    nginx:stable-alpine "docker-entrypoint..." 3 hours ago   Up 3 hours   0.0.0.0:8080->80/tcp, :::8080->80/tcp       cr_nginx
8b31af49c7bd   maildev/maildev "bin/maildev"            3 hours ago   Up 3 hours (healthy) 0.0.0.0:1025->1025/tcp, :::1025->1025/tcp, 0.0.0.0:1080->1080/tcp, :::1080->1080/tcp   cr_mail
6f2ad0ec54a9   redis:alpine   "docker-entrypoint.s..." 3 hours ago   Up 3 hours   0.0.0.0:6379->6379/tcp, :::6379->6379/tcp   cr_redis
s.shrinbel@demo3:~$

```

Рис. 21 – Проверка `docker ps`

2.8.4. Запуск cyberrange-ng.service

Для запуска cyberrange-ng.service (backend сервисы) нужно выполнить следующие команды:

- а) `sudo systemctl start cyberrange-ng.service;`
- б) `sudo systemctl status cyberrange-ng.service.`

Запуск cyberrange-ng.service представлен на Рис. 22.

```

* cyberrange-ng.service - Cyberrange (NG) application
   loaded: loaded (/etc/systemd/system/cyberrange-ng.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2022-12-07 09:52:49 UTC; 2h 47min ago
   Process: 19030 ExecStartPre=/usr/bin/cyberrange-ng/cyberrange stop (code=exited, status=0/SUCCESS)
   Process: 19030 ExecStart=/usr/bin/cyberrange-ng/cyberrange start (code=exited, status=0/SUCCESS)
   Tasks: 961 (limit: 4915)
   CGroup: /system.slice/cyberrange-ng.service
           └─19057 /usr/bin/java -server -XX:+UseStringDeduplication -XX:+OptimizeStringConcat -Xmx256m -Xms128m -XX:MaxMetaspaceSize=512m
           └─19096 /usr/bin/java -server -XX:+UseStringDeduplication -XX:+OptimizeStringConcat -Xmx256m -Xms128m -XX:MaxMetaspaceSize=512m
           └─19138 /usr/bin/java -server -XX:+UseStringDeduplication -XX:+OptimizeStringConcat -Xmx256m -Xms128m -XX:MaxMetaspaceSize=512m
           └─19181 /usr/bin/java -server -XX:+UseStringDeduplication -XX:+OptimizeStringConcat -Xmx512m -Xms128m -XX:MaxMetaspaceSize=512m
           └─19228 /usr/bin/java -server -XX:+UseStringDeduplication -XX:+OptimizeStringConcat -Xmx256m -Xms128m -XX:MaxMetaspaceSize=512m
           └─19272 /usr/bin/java -server -XX:+UseStringDeduplication -XX:+OptimizeStringConcat -Xmx256m -Xms128m -XX:MaxMetaspaceSize=512m
           └─19323 /usr/bin/java -server -XX:+UseStringDeduplication -XX:+OptimizeStringConcat -Xmx1024m -Xms128m -XX:MaxMetaspaceSize=512m
           └─19370 /usr/bin/java -server -XX:+UseStringDeduplication -XX:+OptimizeStringConcat -Xmx512m -Xms128m -XX:MaxMetaspaceSize=512m
           └─19416 /usr/bin/java -server -XX:+UseStringDeduplication -XX:+OptimizeStringConcat -Xmx256m -Xms128m -XX:MaxMetaspaceSize=512m
           └─19461 /usr/bin/java -server -XX:+UseStringDeduplication -XX:+OptimizeStringConcat -Xmx256m -Xms128m -XX:MaxMetaspaceSize=512m
           └─19508 /usr/bin/java -server -XX:+UseStringDeduplication -XX:+OptimizeStringConcat -Xmx256m -Xms128m -XX:MaxMetaspaceSize=512m
           └─19562 /usr/bin/java -server -XX:+UseStringDeduplication -XX:+OptimizeStringConcat -Xmx256m -Xms128m -XX:MaxMetaspaceSize=512m
           └─23113 java -DRunEventId=1 -Xmx256M -Xms256M -server -Dvertx.logger-delegate-factory-class-name=io.vertx.core.Logging.SLF4JLogD
Dec 07 09:52:44 demo3 cyberrange[19030]: message service started: message service is alive (PID: 19228)
Dec 07 09:52:44 demo3 cyberrange[19030]: mission service started: mission service is alive (PID: 19272)
Dec 07 09:52:45 demo3 cyberrange[19030]: event service started: event service is alive (PID: 19323)
Dec 07 09:52:46 demo3 cyberrange[19030]: scoring service started: scoring service is alive (PID: 19370)
Dec 07 09:52:46 demo3 cyberrange[19030]: scoreboard service started: scoreboard service is alive (PID: 19416)
Dec 07 09:52:47 demo3 cyberrange[19030]: tools service started: tools service is alive (PID: 19461)
Dec 07 09:52:48 demo3 cyberrange[19030]: content service started: content service is alive (PID: 19508)
Dec 07 09:52:48 demo3 cyberrange[19030]: test service started: test service is alive (PID: 19562)
Dec 07 09:52:49 demo3 cyberrange[19030]: Container cr_nginx Running
Dec 07 09:52:49 demo3 systemd[1]: Started Cyberrange (NG) application.
lines 1-13/13 (END)

```

Рис. 22 – Процесс запуска cyberrange-ng.service