Installation of Bloodhound CE on Kali linux using docker

Docker

Has a debian package or you can install the latest version from docker

```
sudo apt update
```

```
sudo apt install docker.io
```

Also install docker-compose (orchestrate the installation of various databases for bloodhound)

```
sudo apt install docker-compose
```

Check installation

```
┌──(kali㉿kali)-[~]
└─$ docker --version
Docker version 26.1.5+dfsg1, build a72d7cd
┌──(kali㉿kali)-[~]
└─$ docker-compose --version
Docker Compose version 2.26.1-4
```

Start Docker

```
sudo systemctl enable docker
sudo systemctl start docker
```

Verify its running

```
┌──(kali㉿kali)-[~/Bloodhound]
└─$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

Change docker group privileges

So you don't need sudo to start containers

```
sudo usermod -aG docker $USER
```

make a docker group

```
newgrp docker
id -nG
```

Install Bloodhound CE

specter ops provides a cli-interface (mainly handy for resetting passwords in bloodhound)

See official quickstart here:
[https://bloodhound.specterops.io/get-started/quickstart/community-edition-quickstart]

Make a dir for script files (the real binaries live in a docker container) can be home/kali/Bloodhound or /opt/BloodHound etc.

```
mkdir Bloodhound
```

```
cd Bloodhound
```

Download and install bloodhound-cli

Download:

```
wget https://github.com/SpecterOps/bloodhound-cli/releases/latest/download/bloodhound-cli-linux-amd64.tar.gz
```

Unzip:

```
tar -xvzf bloodhound-cli-linux-amd64.tar.gz
```

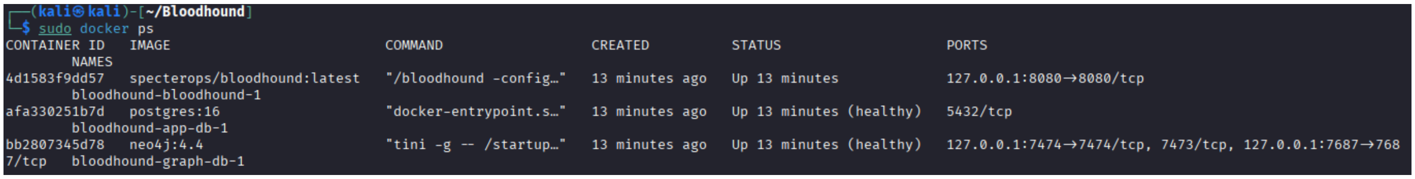
Install:

```
sudo ./bloodhound-cli install
```

Things should now be happening in the shell

```
docker ps
```

And you should see 3 containers running



```
┌──(kali㉿kali)-[~/Bloodhound]
└─$ sudo docker ps
CONTAINER ID   IMAGE                            COMMAND                  CREATED          STATUS                    PORTS
        NAMES
4d1583f9dd57   specterops/bloodhound:latest     "/bloodhound -config…"   13 minutes ago   Up 13 minutes             127.0.0.1:8080->8080/tcp
        bloodhound-bloodhound-1
afa330251b7d   postgres:16                      "docker-entrypoint.s…"   13 minutes ago   Up 13 minutes (healthy)   5432/tcp
        bloodhound-app-db-1
bb2807345d78   neo4j:4.4                        "tini -g -- /startup…"   13 minutes ago   Up 13 minutes (healthy)   127.0.0.1:7474->7474/tcp, 7473/tcp, 127.0.0.1:7687->768
7/tcp   bloodhound-graph-db-1
```

Also check that the cli is working (command also set up things..)

```
./bloodhound-cli check
```

During installation, it also spat out a long complicated password for neo4j in the terminal, by installing the bloodhound-cli, this can easily be reset:

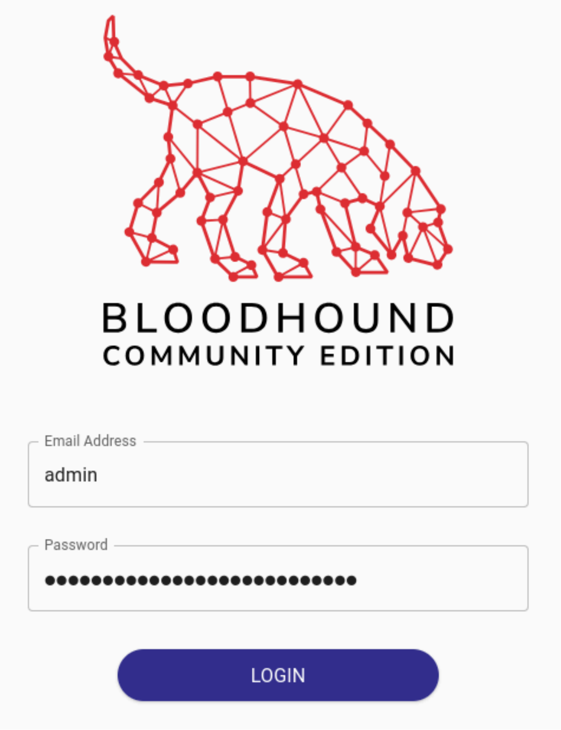
```
./bloodhound-cli resetpwd
```

More commands here:

```
./bloodhound-cli help
```

Initial Bloodhound Setup

The app is hosted on localhost port 8080 login with 'admin' and the long complicated password



Email Address: admin

Password: ●●●●●●●●●●●●●●●●●●●●●●●●●●●●●

You will then be prompted to reset password

Remember the password..

It should now be working

Persistence and easy start

Make a shell script `bloodhound.sh` in the folder you installed bloodhound-cli (fx. ~/Bloodhound)

```
#!/bin/bash

# -------------------------------------
# BloodHound CE Launcher Script
# -------------------------------------
# Usage: bloodhound [--help]
# Default behavior: start
# -------------------------------------

BH_CLI="$HOME/Bloodhound/bloodhound-cli"
BH_USER="admin"
BH_PASS="COPY-PASTE-FireFox PW here"
BH_URL="http://localhost:8080"

# Check if BloodHound is already installed (container exists)
function is_installed() {
    "$BH_CLI" running 2>/dev/null | grep -q -i "container name"
}

function start_bloodhound() {
    echo "[*] Starting BloodHound CE..."
    if ! is_installed; then
        echo "[*] Installing BloodHound CE (first run)..."
        "$BH_CLI" install --yes
    else
        echo "[*] BloodHound is already installed, skipping YAML overwrite."
    fi
    # Start the container
    "$BH_CLI" up
    sleep 5
    echo "[*] Opening BloodHound web UI at $BH_URL"
    xdg-open "$BH_URL"
    echo "[*] Login with username: $BH_USER and password: $BH_PASS"
}

function show_help() {
    cat << EOF
Usage: bloodhound [--help]

Options:
  --help      Show this help message

Default behavior: start BloodHound CE and open web UI at $BH_URL
EOF
}

# Main
case "$1" in
    --help|help)
        show_help
        ;;
    ""|start)
        start_bloodhound
        ;;
    *)
        echo "[!] Unknown option: $1"
        show_help
        exit 1
        ;;
esac
```

Make it executable

```
sudo chmod +x ~/Bloodhound/bloodhound.sh
```

Add Symlink so it will run anywhere

```
sudo ln -sf ~/Bloodhound/bloodhound.sh /usr/local/bin/bloodhound
```

Usage:

```
bloodhound        # start BloodHound CE
bloodhound --help # show help
```

Now you can launch bloodhound ce from anywhere by typing `bloodhound`

Data Ingressors

"Default" are SharpHound anf AzureHound which have community editions

Rusthound

Rust-based rusthound really quick (apparently only does one LDAP query to enumerate the whole AD)

Installation

Requires rust to be installed: [https://rust-lang.org/tools/install/]

```
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

Choose default installation

prereqs (see rusthound):

```
# Debian/Ubuntu
sudo apt-get -y update && sudo apt-get -y install gcc clang libclang-dev libgssapi-krb5-2 libkrb5-dev
libsasl2-modules-gssapi-mit musl-tools gcc-mingw-w64-x86-64
```

install rusthound-ce

```
cargo install rusthound-ce
```

Check it has been installed

```
rusthound-ce --help
```