

Задание №6. Mikrotik

Устанавливаем соединение с Mikrotik

Немного теории про шлюз и туннель

Используем маршрутизатор Mikrotik LTAP Mini LTE Kit



В маршрутизаторе установлена СИМ-карта для доступа маршрутизатора в сеть Интернет. Через сеть Интернет связь с помощью туннеля OpenVPN организована связь до виртуального маршрутизатора OPNsense, который обеспечивает связь маршрутизатора Mikrotik с вашим виртуальным сервером.

Схема:



Проверка связности с Mikrotik

IP маршрутизатора: 192.168.111.2

Сделаем Ping до маршрутизатора:

```
Ping 192.168.111.2
```

Если в ответе увидим

```
64 bytes from 192.168.111.2: icmp_seq=1 ttl=64 time=0.072 ms
```

то связь с маршрутизатором есть.

Если ответов не будет, значит связи с маршрутизатором нет.

Прервём выполнение команды ping нажатием Ctrl+C

Для удаленного доступа на маршрутизатор используем программу ssh:

```
Ssh -l user1 192.168.111.2 interface wireless registration-table print
```

Эта команда задаёт имя пользователя = user1 и передает маршрутизатору команду получения списка пользователей WiFi:

```
interface wireless registration-table print
```

Введем пароль: mikrotik

Получим таблицу MAC-адресов пользователей WiFi с уровнями сигналов.:

Подключим свой мобильный телефон к wifi на этом маршрутизаторе:

SSID: Mikrotik, пароль: mikrotik

И снова запустим команду. Список пользователей должен измениться.

Пишем скрипт получения данных с Mikrotik

Будем использовать библиотеку librouteros. Эту библиотеку нужно установить: Обновляем список пакетов:

```
Sudo apt update
```

Проверяем наличие пакета: Apt list | grep python3-librouteros Устанавливаем библиотеку

```
Sudo apt install python3- librouteros
```

Ещё раз проверяем наличие пакета.

Для подключения библиотеки к нашему скрипту в начале скрипта Python напишем: `from librouteros import connect` Описываем соединение: `api = connect(host='192.168.111.2',username='user1',password='mikrotik')` Выполняем команду на маршрутизаторе:

```
wifiuser=api.path('interface', 'wireless', 'registration-table')
```

Затем в цикле `for` проходим по всем полученным записям и отображаем их на экране. В конце скрипта закрываем соединение с роутером.

Скрипт:

```
from librouteros import connect
api = connect(host='192.168.111.2',username='user1',password='mikrotik')
wifiuser=api.path('interface', 'wireless', 'registration-table')
for item in wifiuser:
    print (item, "\n")
api.close()
```

Получаем массив данных о пользователях Wifi сети Mikrotik на нашем роутере. Элементы массива разделены на наименование элемента и значение элемента, разделенные знаком «:». Например, первый элемент массива — это номер строки `'id': '*1'`. Второй элемент массива — это интерфейс `'interface': 'wlan1'` Нас интересует третий элемент массива. Для его отображения в команде `print` напишем:

```
print (item['mac-address'], "\n")
```

Запустите скрипт. Теперь вместо всех параметров нам будут отображаться только MAC-адреса пользователей WiFi сети. Добавим второй параметр – уровень сигнала, отделим его от MAC-адреса разделителем « - »:

```
print (item['mac-address'], " - ", item['signal-strength-ch1'], "\n")
```

Запустите скрипт. Вы получили список MAC-адресов пользователей сети WiFi и их уровни сигналов. Записываем полученные данные в БД SQL В предыдущих заданиях мы в базе данных `pythondb` создали таблицу `registrations` со столбцами `mac` и `signal`. Запишем полученные с роутера значения в эту таблицу. Добавляем в начало скрипта команды работы с SQL:

```
import sqlite3
con = sqlite3.connect('/home/bsk23-01-student-1/python/pythondb')
cur = con.cursor()
```

Затем внутри цикла сформируем строку вставки данных в таблицу SQL:

```
Insert = "INSERT INTO registrations (mac, signal) VALUES ('" + item['mac-address'] + "', '" + item['signal-strength-ch1'] + "')
```

Для проверки выведем получившуюся команду на экран:

```
Print(insert)
```

Выполним скрипт, проверим правильно ли сформирована команда вставки данных. Вставка данных в таблицу SQL выполняется командами:

```
Cur.execute(insert)
Con.commit()
```

В конце скрипта добавим:

```
Cur.close()
```

Выполним скрипт и проверим наличие данных в таблице SQL:

```
Sqlite3 pythondb
Select * from registrations
```

Видим, что новые записи добавляются в дополнение ранее добавленным. Для того, чтобы получить в таблице только актуальные данные нужно предварительно очистить таблицу registrations SQL-командой:

```
DELETE FROM registrations
```

Напишем в скрипте перед циклом:

```
Cur.execute("DELETE FROM registrations")
Con.commit()
```

Итоговый скрипт from librouteros import connect import sqlite3

```
con = sqlite3.connect('/home/bsk23-01-student-1/python/pythondb')
```

```
cur = con.cursor() api = connect(host='192.168.111.2',username='user1',password='mikrotik')
wifuser=api.path('interface', 'wireless', 'registration-table') Cur.execute("DELETE FROM
registrations")
```

```
Con.commit()
```

```
for item in wifuser Insert = "INSERT INTO registrations (mac, signal) VALUES ('" + item['mac-
address'] + "','" + item['signal-strength-ch1'] + "')" Cur.execute(insert)
```

```
Con.commit()
```

```
api.close()
```

Теперь наш сайт показывает данные о пользователях Написанный вами на самостоятельной работе скрипт отображает список пользователей WiFi сети из базы данных. Данные в базе данных обновляются скриптом написанным выше. Запуск скрипта автоматически по расписанию в Linux Xxx Можно сделать чтение списка пользователей и его отображение без записи в БД SQL. Может возникнуть проблема с одновременным доступом на маршрутизатор

From:

<https://sibgu-itlab-wiki.data-pool.ru/> - **SIBGU-ITLAB-WIKI**

Permanent link:

https://sibgu-itlab-wiki.data-pool.ru/zadanie_6._mikrotik?rev=1713382077

Last update: **2024/04/17 19:27**

