Настройка туннелей GRE и IPIP на роутерах Mikrotik

interface31.ru/tech_it/2021/07/nastroyka-tunneley-gre-i-ipip-na-routerah-mikrotik.html

Одна из наиболее часто решаемых системным администратором задач - объединение нескольких сетей в единое пространство, для обеспечения совместной работы с общими ресурсами (site-to-site). Обычно для этих целей используется VPN, тип которого большой роли не играет. Но именно для данной задачи более предпочтительно использовать IPIP или GRE-туннели, особенно если вам требуется хорошая пропускная способность соединения. В данной статье мы расскажем об особенностях настройки и использования данного вида подключений.

Онлайн-курс по MikroTik
Научиться настраивать MikroTik с нуля или систематизировать уже имеющиеся знания можно на углубленном курсе по администрированию MikroTik. Автор курса, сертифицированный тренер MikroTik Дмитрий Скоромнов, лично проверяет лабораторные работы и контролирует прогресс каждого своего студента. В три раза больше информации, чем в вендорской программе MTCNA, более 20 часов практики и доступ навсегда.

Сначала коротко о протоколах. **GRE** (*Generic Routing Encapsulation - общая инкапсуляция маршрутов*) - протокол инкапсуляции, разработан компанией Cisco и предназначен для инкапсуляции пакетов сетевого уровня (L3) в IP-пакеты. **IPIP** (*IP Encapsulation within IP - инкапсуляция IP в IP*) во многом похож на GRE, но работает только с **IPv4-трафиком**. Наиболее популярным и используемым протоколом является GRE, его поддержка присутствует во всех современных ОС и сетевом оборудовании. Mikrotik поддерживает оба вида туннелей.

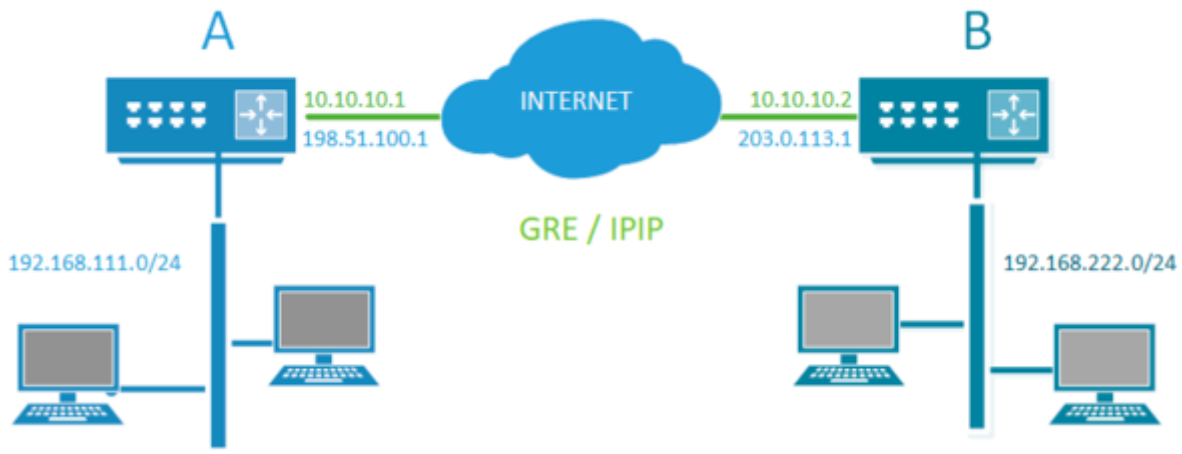
Туннели, созданные с помощью данных протоколов, не имеют никаких механизмов обеспечения безопасности (шифрование, аутентификация), поэтому в чистом виде они практически не используются. Для обеспечения нужного уровня безопасности используется **IPsec**, поверх которого уже разворачивается GRE или IPIP-туннель (*GRE over IPsec, IPIP over IPsec*). Далее, говоря о данном типе туннелей мы будем подразумевать ввиду именно их.

Еще одна особенность указанных протоколов - они работают **без сохранения состояния соединения** (*stateless*) и понять в каком состоянии находится туннель невозможно. Мы можем только настроить обе стороны и проверить передачу данных между ними. Кроме очевидных минусов такое решение имеет и свои плюсы, GRE или IPIP-интерфейсы являются статичными и присутствуют в системе вне зависимости от состояния туннелей, что облегчает настройку маршрутизации. А состояние туннеля позволяют контролировать механизмы RouterOS, которые с заданной периодичностью умеют проверять доступность его второго конца.

Ни GRE, ни IPIP **не используют порты**, поэтому они не могут преодолеть NAT, это требует от обоих узлов иметь выделенные IP-адреса или находиться в одной сети. Проблема NAT частично снимается при использовании IPsec, за счет использования протокола NAT-T, но требование выделенных адресов узлов остается. Кроме того, по этой причине вы не сможете установить более одного GRE или IPIP-соединения между узлами.

Итак, подведем коротко итог: для использования GRE или IPIP-туннелей вам потребуются **выделенные IP-адреса с обоих сторон** и для защиты передаваемых данных **обязательно использовать IPsec**. Что касается оборудования, то предпочтительно использовать роутеры с аппаратной поддержкой шифрования - hEX, RB3011/4011 и все остальные модели на базе процессоров ARM. В этом случае вполне достижима пропускная способность туннеля на уровне 300-400 МБит/с. На остальных моделях роутеров (MIPSBE, SMIPS) вы получите не более 30-40 МБит/с. Подробнее об этом вы можете прочитать здесь.

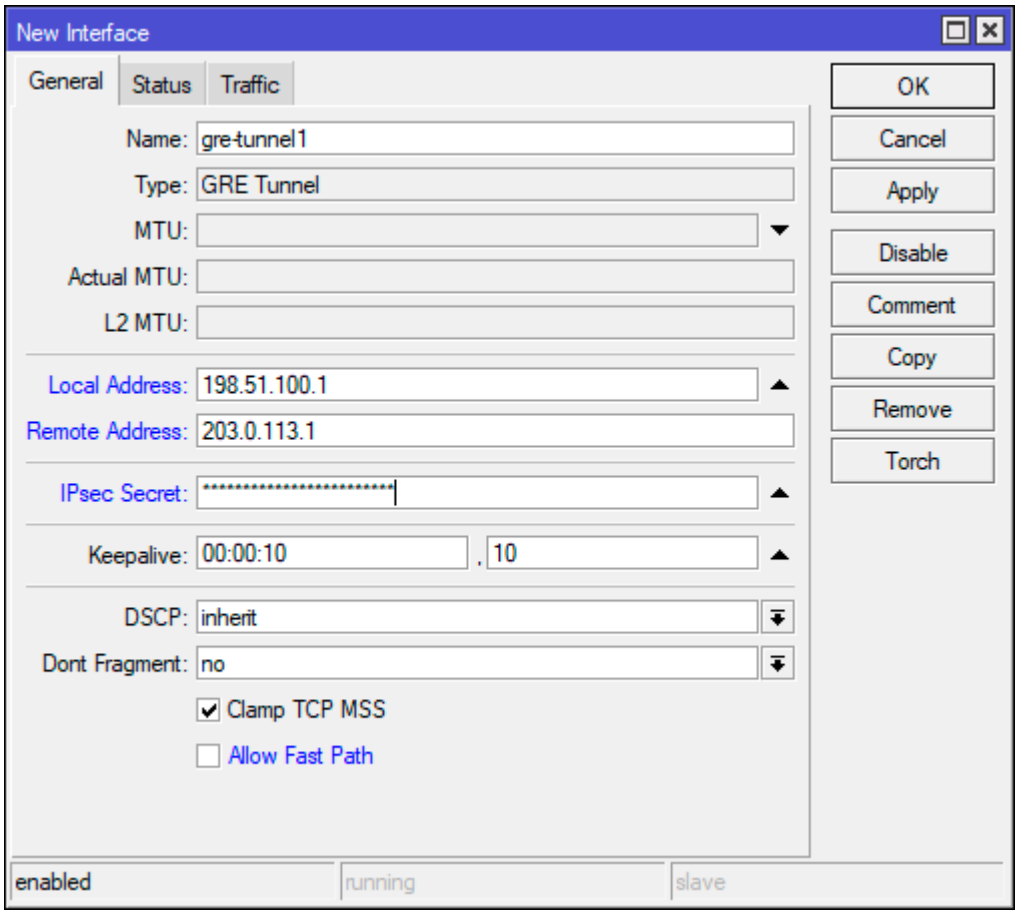
Далее мы будем придерживаться следующей схемы:



Согласно которой у нас имеются две условные сети: **A** - 192.168.111.0/24, внешний IP-адрес 198.51.100.1 и **B** - 192.168.222.0/24, внешний адрес 203.0.113.1. Между ними мы будем поднимать GRE или IPIP-туннель с внутренними адресами 10.10.10.1 и 10.10.10.2.

Настройка GRE-туннеля

Открываем Winbox и переходим в **Interfaces - Interface** где добавляем новый интерфейс с типом **GRE Tunnel**, в открывшемся окне заполняем поля: **Local Address** - внешний IP-адрес этого роутера, **Remote Address** - внешний IP-адрес противоположного роутера, **IPsec Secret** - общий ключ IPsec, рекомендуется использовать длинную случайную строку из цифр, букв в обоих регистрах и спецсимволов. Также обязательно снимите флаг **Allow Fast Path**.

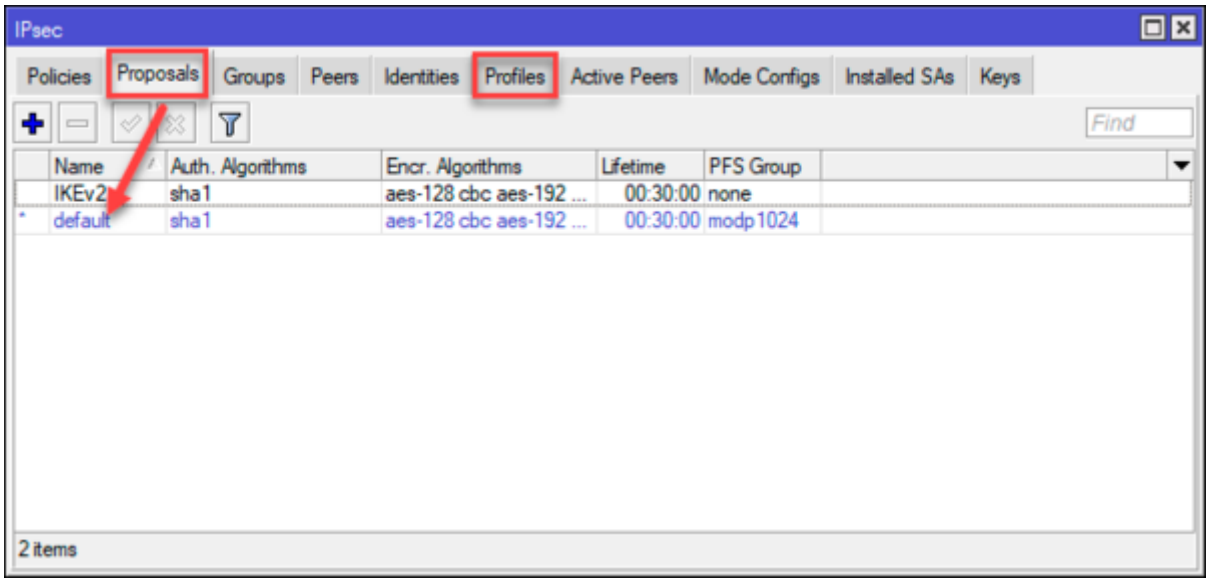


В терминале это можно выполнить командой:

```
/interface gre
add allow-fast-path=no ipsec-secret=MyIP$ecret123 local-address=198.51.100.1
name=gre-tunnel1 remote-address=203.0.113.1
```

Полностью аналогичную настройку выполняем и на втором роутере, только меняем местами **Local Address** и **Remote Address**, после чего туннель должен перейти в рабочее состояние. За отслеживание состояние туннеля отвечает параметр **Keepalive**, по умолчанию он предполагает десять попыток с интервалов в 10 секунд, если за это время с противоположной стороны не будет получен ответ, то туннель будет считаться неработоспособным.

Важный момент связан с настройками IPsec, RouterOS использует для туннелей настройки по умолчанию и нет возможности это переопределить, поэтому на обоих роутерах профили **default** в **IP - IPsec - Proposals** и **Profiles** должны иметь одинаковые настройки.

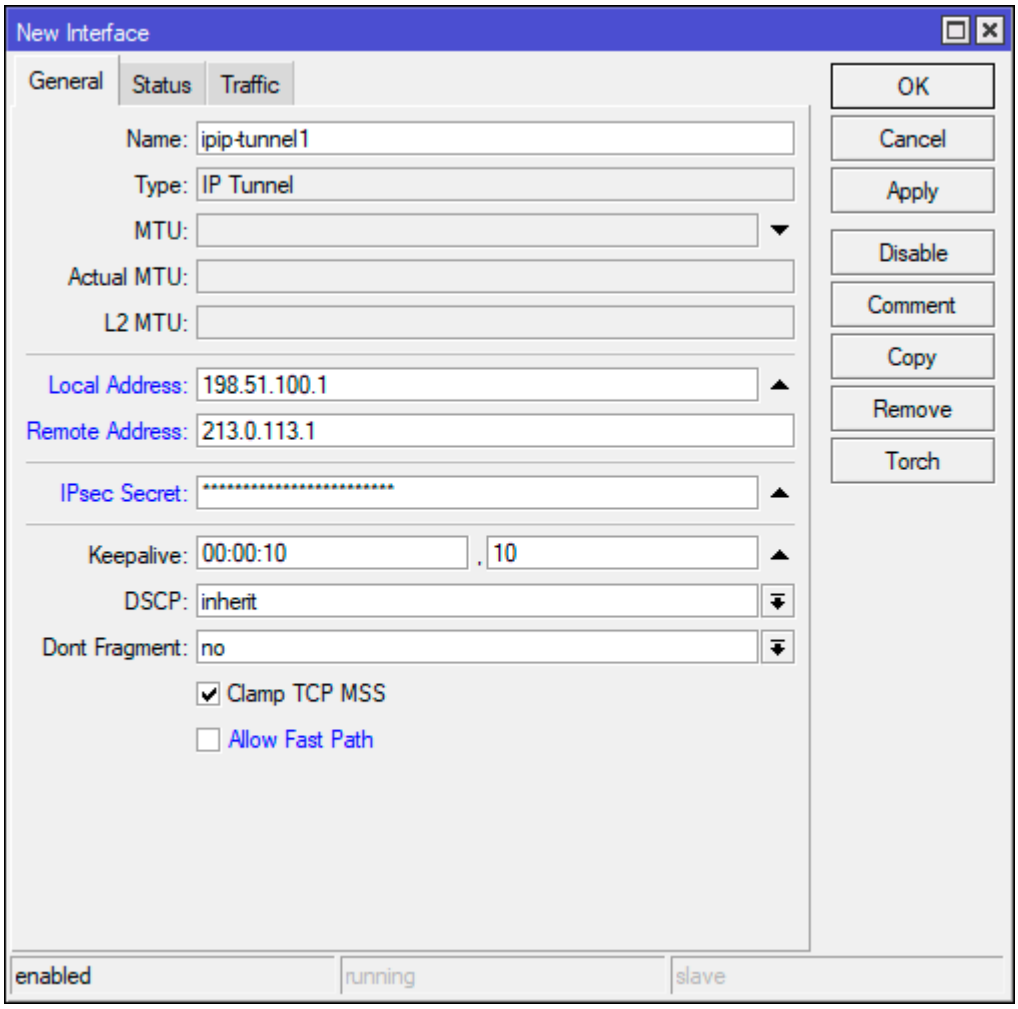


В противном случае вы будете получать ошибку при согласовании параметров IPsec:

Если все сделано правильно, то в Interfaces - Interface напротив туннеля появится флаг **R - running**, что означает, что туннель находится в рабочем состоянии.

Настройка IPIP-туннеля

Настройка данного вида туннеля ничем не отличается от GRE, также переходим в **Interfaces - Interface** и добавляем новый интерфейс с типом **IP Tunnel**. Указываем все те же параметры: **Local Address** - внешний IP-адрес этого роутера, **Remote Address** - внешний IP-адрес противоположного роутера, **IPsec Secret** - общий ключ IPsec, также снимаем флаг **Allow Fast Path**.



В терминале это же действие:

```
/interface ipip
add allow-fast-path=no ipsec-secret=MyIP$ecret123 local-address=198.51.100.1
name=ipip-tunnel1 remote-address=203.0.113.1
```

Затем дублируем настройки на второй роутер, заменяя местами **Local Address** и **Remote Address**, также учитываем все то, что было сказано выше о настройках IPsec.

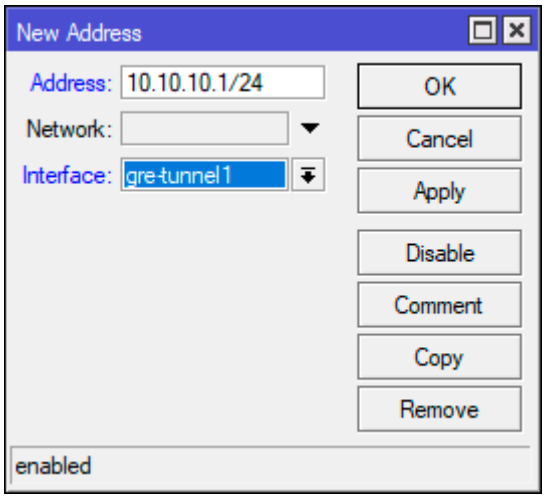
Настройка маршрутизации

Итак, туннель поднят, теперь нужно пустить в него трафик между сетями. Прежде всего присвоим адреса туннельным интерфейсам. Согласно схеме со стороны роутера A это будет 10.10.10.1, а со стороны роутера B - 10.10.10.2. Переходим в **IP - Addresses** и добавляем новый адрес: **Address - 10.10.10.1/24** - именно так, с указанием префикса (/24, что соответствует маске 255.255.255.0), в противном случае сеть у вас работать не будет. В поле **Interface** указываем имя **интерфейса GRE или IPIP-туннеля**.

В терминале для этого же действия выполните команду:

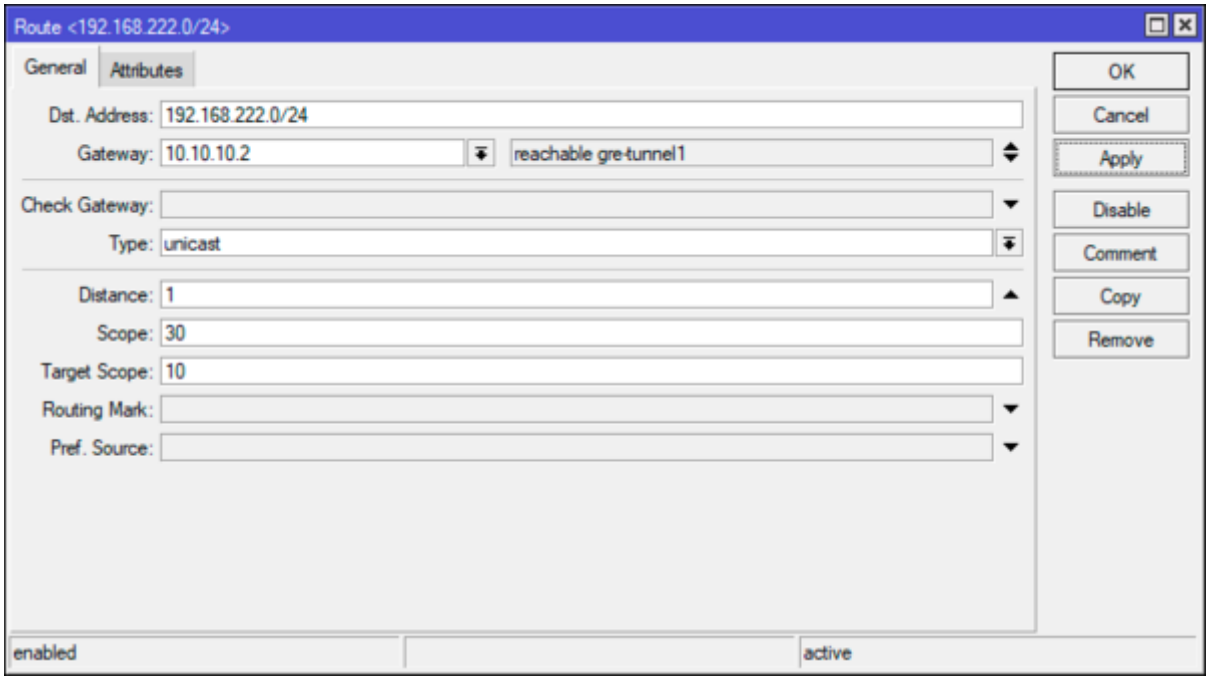
```
/ip address
add address=10.10.10.1/24 interface=gre-tunnel1 network=10.10.10.0
```

Где вместо **interface=gre-tunnel1** укажите имя собственного туннельного интерфейса.



Аналогичные настройки следует выполнить и на втором роутере.

Теперь приступим к указанию маршрутов, для роутера А нам нужно указать маршрут к сети 192.168.222.0/24 через туннель. Переходим в **IP - Routes** и создаем новый маршрут. В качестве **Dst. Address** указываем сеть назначения - 192.168.222.0/24, в поле **Gateway** указываем шлюз в эту сеть - противоположный конец туннеля, который имеет адрес 10.10.10.2, после того как мы нажмем **Apply** в поле рядом со шлюзом появится исходящий интерфейс, в качестве которого будет выступать наш туннель.

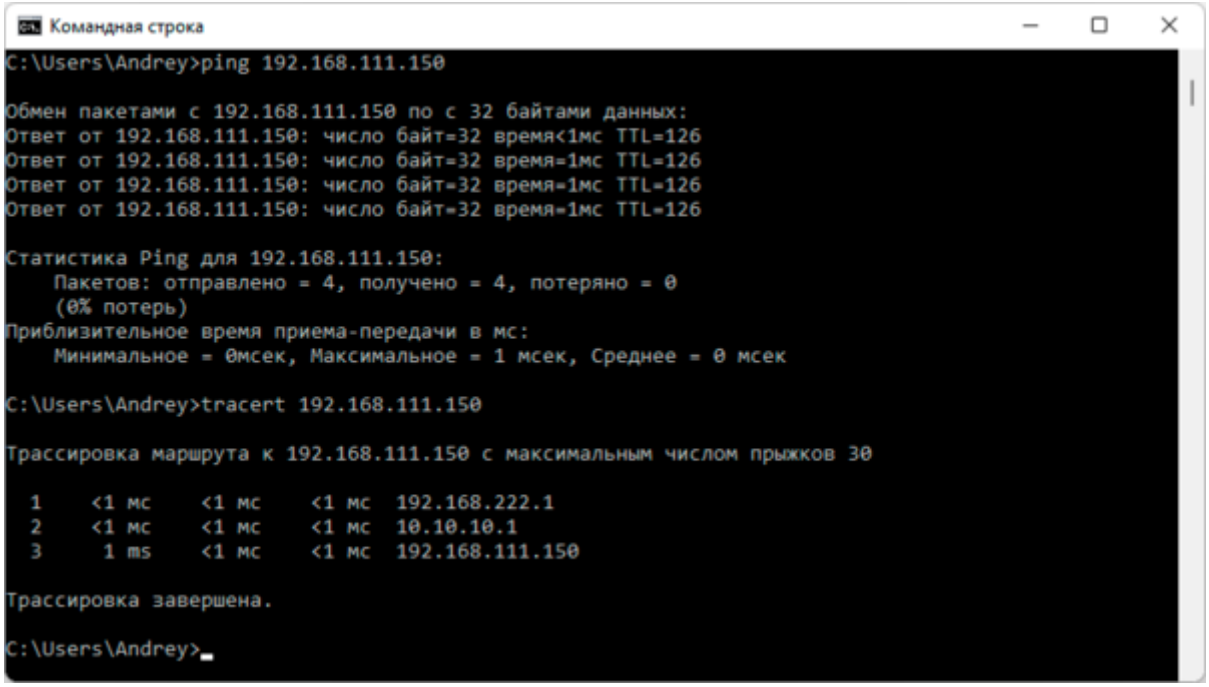


В терминале:

```
/ip route
add distance=1 dst-address=192.168.222.0/24 gateway=10.10.10.2
```

На втором роутере делаем аналогичные настройки с учетом IP-адреса роутера и сети назначения.

После чего пробуем получить из одной сети доступ к узлу другой. Если все сделано правильно, то попытка увенчается успехом.



Онлайн-курс по MikroTik

Научиться настраивать MikroTik с нуля или систематизировать уже имеющиеся знания можно на углубленном курсе по администрированию MikroTik. Автор курса, сертифицированный тренер MikroTik Дмитрий Скоромнов, лично проверяет лабораторные работы и контролирует прогресс каждого своего студента. В три раза больше информации, чем в вендорской программе MTCNA, более 20 часов практики и доступ навсегда.