УДК 004.056:681.3.06

Олександр Яценко,
асистент кафедри комп’ютерних наук та інформаційних технологій
Оксана Яценко,
асистент кафедри комп’ютерних наук та інформаційних технологій
Житомирський державний університет імені Івана Франка

НАЛАШТУВАННЯ ПРОТОКОЛУ WIREGUARD НА ОБЛАДНАННІ ФІРМИ MIKROTIK

Зростаюча цифровізація суспільства та бізнесу висуває підвищені вимоги до забезпечення кібербезпеки та конфіденційності даних. Віртуальні приватні мережі (VPN) є основним інструментом для створення захищених каналів зв'язку через недовірені мережі, такі як Інтернет. Проте традиційні VPN-протоколи, зокрема OpenVPN та IPsec, часто характеризуються значними накладними витратами, складністю конфігурації та потенційними вразливостями, пов'язаними з великим обсягом коду.

Обладнання фірми MikroTik, що працює під управлінням операційної системи RouterOS, широко використовується в мережевих інфраструктурах по всьому світу завдяки своїй функціональності, гнучкості та економічній ефективності. Інтеграція підтримки WireGuard безпосередньо в RouterOS відкриває нові можливості для розгортання високопродуктивних та безпечних VPN-рішень на цій платформі.

WireGuard – це відносно новий VPN-протокол, який здобув значну популярність завдяки своїй простоті, високій швидкості та надійним криптографічним алгоритмом. Його архітектура мінімізує обсяг коду, що знижує ймовірність помилок та атак, а також дозволяє досягти продуктивності, порівнянної з власними мережевими інтерфейсами.

WireGuard – безкоштовне програмне забезпечення з відкритим вихідним кодом, яке реалізує зашифровані віртуальні приватні мережі (VPN). Він був розроблений для простоти використання технології VPN, високої продуктивності та низької поверхні атаки [2]. WireGuard прагне до кращої продуктивності та більшої потужності, ніж IPsec та OpenVPN, два інших поширених протоколи тунелювання [3]. Протокол WireGuard передає трафік за протоколом UDP [4].

WireGuard базується на принципах сучасних криптографічних алгоритмів та мінімалізму [2]. Його ключовими особливостями є:

1. Сучасна криптографія: він використовує протокол обміну ключами Noise Protocol Framework, алгоритм Діффі-Хеллмана на еліптичних кривих Curve25519 для обміну ключами, потоковий шифр ChaCha20 для шифрування даних та функцію імітостійкості Poly1305 для автентифікації, а це забезпечує високий рівень безпеки та стійкість до сучасних криптографічних атак.

2. Спрощений обмін ключами: замість складних сертифікатів, WireGuard використовує публічні та приватні ключі для ідентифікації пірів, що значно спрощує початкове налаштування та подальше керування.

3. Стійкість до зміни IP-адрес: WireGuard підтримує «роумінг», що дозволяє клієнтам змінювати IP-адреси без розриву з'єднання, що є перевагою для мобільних пристроїв.

4. Висока продуктивність: мінімальний обсяг коду та оптимізовані алгоритми дозволяють WireGuard працювати з високою пропускною здатністю та низькими затримками.

Перед початком налаштування потрібно переконатися, що встановлена сьома версія RouterOS (рекомендується 7.19.1 або новіша) та є доступ до пристрою MikroTik за допомогою WinBox або SSH/Telnet (командного рядка).

Загалом налаштування WireGuard на MikroTik включає такі етапи: генерацію ключів, налаштування WireGuard сервера (створення інтерфейсу WireGuard, налаштування пірів та конфігурація IP-адрес), налаштування маршрутизації та фаєрволу. Розглянемо приклад конфігурації для сценарію «Сервер-Клієнт».

Етап 1. Генерація ключів. Кожен пір (сервер або клієнт) у WireGuard потребує пари публічного та приватного ключів. Приватний ключ зберігається в секреті, а публічний обмінюється між пірами.

Фрагмент коду для генерації ключів на MikroTik `/interface wireguard keys generate` виведе приватний та публічний ключі, які потрібно зберегти.

Приклад виводу:

```
private-key: <генерується приватний ключ>
```

```
public-key: <генерується публічний ключ>
```

Цю процедуру потрібно повторити для кожного клієнта, який буде підключатися до сервера.

Для генерації ключів також можна використати зовнішні інструменти для генерації ключів (наприклад, wg genkey на Linux або wgcf для Windows).

Етап 2. Налаштування WireGuard сервера на MikroTik

Фрагмент коду для Створення WireGuard інтерфейсу:

```
/interface wireguard add name=wg-server listen-port=51820 private-key="<приватний ключ сервера>"
```

Де:

`name=wg-server`: назва інтерфейсу WireGuard;

`listen-port=51820`: порт, на якому сервер WireGuard буде очікувати вхідні з'єднання (можна змінити);

`private-key="<приватний ключ сервера>"`: приватний ключ, згенерований для сервера.

На цьому ж етапі призначається IP-адреса WireGuard інтерфейсу. Для цього потрібно створити віртуальну IP-мережу для тунелю WireGuard. Наприклад, 10.0.0.1/24 для сервера. Код буде мати вигляд:

```
/ip address add address=10.0.0.1/24 interface=wg-server
```

Крім того для кожного клієнта необхідно додати запис про пір на сервері:

```
/interface wireguard peers add interface=wg-server public-key="<публічний ключ клієнта>" allowed-address="10.0.0.2/32"
```

Де:

`interface=wg-server`: вказує, до якого WireGuard інтерфейсу додається пір;

`public-key="<публічний ключ клієнта>"`: публічний ключ відповідного клієнта;

`allowed-address="10.0.0.2/32"`: IP-адреса, яку WireGuard сервер дозволить клієнту використовувати в тунелі. Це також впливає на маршрутизацію: трафік до цієї адреси буде направлятися до цього піра. Для доступу до всієї внутрішньої мережі клієнтів можна вказати 10.0.0.0/24 або інший діапазон.

Етап 3. Налаштування WireGuard клієнта

Клієнт WireGuard – це роутер MikroTik з динамічною (сірою) IP-адресою. Спочатку потрібно створити інтерфейс WireGuard на боці клієнта. Припустимо, клієнт працює на Linux, Windows, macOS або мобільному пристрої. Конфігураційний файл клієнта (wg0.conf або аналогічний) матиме вигляд:

```
Ini, TOML
[Interface]
PrivateKey = <приватний ключ клієнта>
Address = 10.0.0.2/32
DNS = 8.8.8.8 # Опціонально, DNS-сервер для клієнта

[Peer]
PublicKey = <публічний ключ сервера MikroTik>
Endpoint = <зовнішня IP-адреса MikroTik сервера>:<порт WireGuard сервера>
AllowedIPs = 0.0.0.0/0 # Для маршрутизації всього трафіку через VPN
#PersistentKeepalive = 25 # Опціонально, для підтримки з'єднання через NAT
```

Де:

`PrivateKey`: приватний ключ клієнта.

`Address`: IP-адреса клієнта всередині віртуальної мережі WireGuard.

`PublicKey`: публічний ключ сервера MikroTik.

`Endpoint`: зовнішня IP-адреса та порт WireGuard сервера MikroTik.

`AllowedIPs`: вказує, які IP-адреси мають маршрутизуватися через тунель. `0.0.0.0/0` означає весь трафік (повне тунелювання). Якщо потрібно лише доступ до локальної мережі, потрібно вказати діапазон локальної мережі (наприклад, 192.168.1.0/24).

Етап 3. Налаштування маршрутизації та фаєрволу на MikroTik (сервер)

Для коректної роботи VPN необхідно налаштувати маршрутизацію та правила фаєрволу на сервері MikroTik.

Для цього потрібно дозволити вхідні UDP-з’єднання на порт WireGuard сервера (за замовчуванням 51820):

```
/ip firewall filter add chain=input proto=udp dst-port=51820 action=accept comment="Allow WireGuard"
```

Якщо клієнти WireGuard мають отримати доступ до Інтернету через сервер MikroTik, необхідно налаштувати NAT:

```
/ip firewall nat add chain=srcnat action=masquerade out-interface=<інтерфейс, через який MikroTik виходить в Інтернет> src-address=10.0.0.0/24 comment="WireGuard NAT"
```

Де:

`<інтерфейс, через який MikroTik виходить в Інтернет>`: назва інтерфейсу, підключеного до провайдера (наприклад, ether1, pppoe-out1);

`src-address=10.0.0.0/24`: мережа, яка використовується для WireGuard тунелю.

Якщо за сервером MikroTik є внутрішні локальні мережі, до яких клієнти WireGuard повинні мати доступ, то потрібно переконатись, що маршрути до цих мереж відомі WireGuard інтерфейсу або що NAT коректно працює для цих напрямків. Зазвичай, якщо використовується `action=masquerade` з `src-address=10.0.0.0/24` на `out-interface` то до локальної мережі, окрема маршрутизація може не знадобитися, але це залежить від топології.

Після налаштування потрібно перевірити стан WireGuard з'єднань на MikroTik та на клієнті (Linux). Фрагмент коду для MikroTik:

```
/interface wireguard peers print
```

покаже список пірів, їхній статус, останню активність та обсяг переданих даних.

На клієнті (Linux):

```
Bash
sudo wg show
```

Аналогічно, можна відобразити інформацію про WireGuard інтерфейс та піри.

Детальне викладення процесу налаштування, від генерації ключів до конфігурації фаєрволу, демонструє відносну легкість впровадження WireGuard на платформі RouterOS. Це дозволяє фахівцям з мережевих технологій використовувати переваги обох компонентів для створення безпечних, швидких та масштабованих мережевих рішень, що є критично важливим в умовах постійного зростання кіберзагроз та потреби в гнучкій інфраструктурі. Подальші дослідження можуть бути зосереджені на оптимізації продуктивності WireGuard

в умовах високого навантаження та інтеграції з централізованими системами управління і моніторингу.

Список використаних джерел та літератури

5. MikroTik. *MikroTik Routers and Wireless*. URL: https://mikrotik.com/ (date of access: 29.05.2025).

6. Jason A. Donenfeld. WireGuard: fast, modern, secure VPN tunnel. *www.wireguard.com*. URL: https://www.wireguard.com/ (date of access: 29.05.2025).

7. Applied Cryptography and Network Security / ed. by B. Preneel, F. Vercauteren. Cham: Springer International Publishing, 2018. URL: https://doi.org/10.1007/978-3-319-93387-0 (date of access: 29.05.2025)/

8. Known Limitations – WireGuard. *WireGuard: fast, modern, secure VPN tunnel*. URL: https://www.wireguard.com/known-limitations/ (date of access: 29.05.2025).