Лабораторна робота № 12
НАЛАГОДЖЕННЯ ТА ДОСЛІДЖЕННЯ ЗАСОБІВ ВІДДАЛЕНОГО ДОСТУПУ ТА АДМІНІСТРУВАННЯ

Мета заняття: ознайомитися з особливостями функціонування протоколів та засобів віддаленого доступу та адміністрування; отримати практичні навички налагодження, моніторингу та діагностування засобів віддаленого доступу та адміністрування сучасних ОС; дослідити можливості ОС Windows, Linux, Cisco IOS з організації, налагодження та функціонування незахищених та захищених віддалених мережних підключень на базі протоколів Telnet та SSH.

Теоретичні відомості

Протоколи віддаленого доступу

Надзвичайно важливим питанням системного та мережного адміністрування є забезпечення постійного доступу до комунікаційних пристроїв та кінцевих вузлів мережі. Виконання цього завдання у сучасних мережах забезпечують так звані протоколи віддаленого доступу. Протокол віддаленого доступу забезпечує доступ адміністратора/користувача з одного вузла до іншого через наявну мережну інфраструктуру. Як правило, такі протоколи побудовані за клієнт-серверною схемою. До протоколів віддаленого доступу належать протоколи:

– Telnet, TELecommunication NETwork;
– SSH, Secure SHell;
– RLOGIN, Remote LOGIN;
– RDP, Remote Desktop Protocol;
– RFB, Remote FrameBuffer.

Для цих протоколів розроблено ряд інструментальних засобів для реалізації віддаленого доступу – програм-серверів та програм термінальних клієнтів. У багатьох ОС термінальні клієнти є вбудованими. Більшість із вищеперерахованих протоколів (зокрема, Telnet та SSH) орієнтовані на використання інтерфейсу командного рядка, лише деякі (зокрема, RDP) – на використання графічних засобів. Найбільш поширеними протоколами сьогодні є протоколи віддаленого доступу Telnet та SSH. Протокол Telnet є недостатньо захищеним, тому у практиці адміністрування рекомендується застосовувати засоби, які базуються на протоколі SSH.

Протокол віддаленого доступу Telnet

Telnet (англ. TELecommunication NETwork, телекомунікаційна мережа) – протокол, який був розроблений одним із перших у стеку TCP/IP. Перші згадки про нього наявні у стандарті RFC-15 „Network Subsystem for Time Sharing Hosts", що був випущений у 1969 р. До 1983 р., коли було випущено основну специфікацію протоколу RFC-854 „Telnet Protocol Specification", розроблено й опубліковано понад 20 стандартів RFC, які покращували і розширювали можливості протоколу. Останній стандарт, який має відношення до протоколу Telnet, – це стандарт RFC-5198 „Unicode Format for Network Interchange", випущений у 2008 р.

Необхідність розробки протоколу Telnet була зумовлена потребою спрощення підключення до віддалених вузлів та пристроїв різних типів. У повсякденній діяльності використовується велика кількість різнотипних комп'ютерів, кожен із яких потребує сумісного обладнання для введення-виведення інформації (рис. 1, *а*), і це є проблемою.

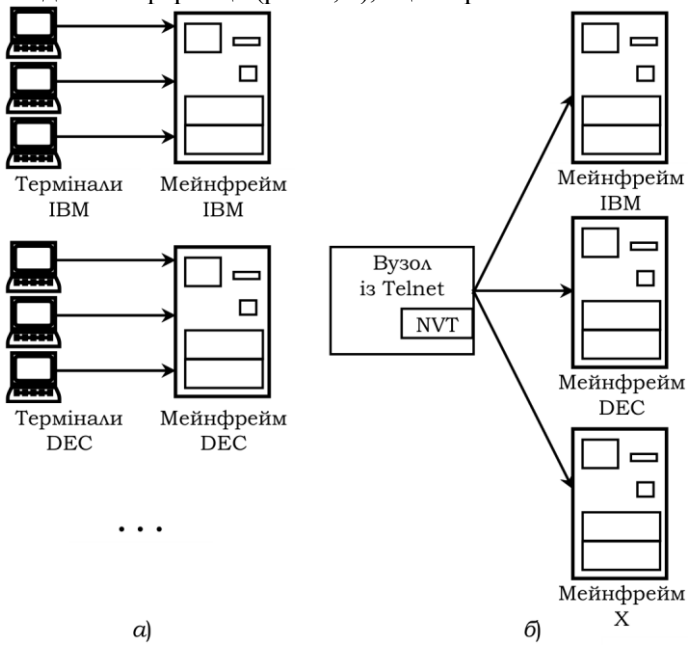


Рис. 1. Концепція віртуального термінала

Ситуацію ускладнює і те, що на цих комп'ютерах використовуються різні ОС, різні таблиці кодування та різне програмне забезпечення. Тому виникла потреба у службі емуляції термінала, яка б замінила спеціалізовані пристрої та програми однією службою. Це було реалізовано за рахунок концепції віртуального термінала (NVT, Network Virtual Terminal) (див. рис. 1, *б*). Віртуальний термінал отримує дані, які вводяться у клієнтській системі, і перекладає їх на „універсальну мову". Отримані дані перекладаються з „універсальної мови" на спеціалізовану мову, яка сприймається вузлом. Це дає змогу будь-якому спеціалізованому клієнтові взаємодіяти з будь-яким спеціалізованим сервером.

Telnet є клієнт-серверним протоколом. Належить цей протокол до прикладного рівня моделі OSI та прикладного рівня стеку TCP/IP. Для передачі своїх повідомлень Telnet використовує засоби надійного транспортного протоколу TCP. Саме TCP забезпечує стабільний і надійний зв'язок. За замовчуванням сервер Telnet застосовує порт 23. Клієнт Telnet для організації обміну обирає вільний порт із діапазону динамічних портів системи.

Клієнт Telnet може бути налагоджений на підключення до іншого порту сервера, на якому працює інша служба. Це дозволяє використовувати клієнт Telnet для передачі команд та отримання відповідей на команди конкретним службам додатків та для потреб діагностики.

Telnet може працювати у таких режимах:

– напівдуплексний режим;

– посимвольний режим;

– рядковий режим;

– локальний режим.

Напівдуплексний режим уважається застарілим і у сучасних системах не застосовується. У посимвольному режимі кожен уведений символ відразу ж передається вузлу для обробки, а потім повертається клієнтові. У низькошвидкісних мережах це створює затримки. У багатьох реалізаціях згаданий режим застосовується за замовчуванням. У рядковому режимі текст команди спочатку виводиться на екран, і лише закінчені рядки передаються віддаленому вузлу для обробки. У локальному режимі обробка символів проводиться у локальній системі під контролем віддаленої системи.

Протокол віддаленого доступу SSH

Протокол SSH (англ. Secure SHell, „безпечна оболонка") – це мережний протокол віддаленого доступу, який дає змогу здійснювати

віддалене управління операційною системою будь-якого мережного пристрою і безпечно передавати у незахищеному середовищі повідомлення будь-якого іншого мережного протоколу (наприклад, здійснювати тунелювання TCP-з'єднань для передачі файлів). За функціональністю схожий на протоколи Telnet, rlogin та rsh (два останні нині не застосовуються), але на відміну від них, шифрує увесь трафік, що передається, зокрема і паролі. Окрім шифрування може також здійснювати стиснення даних. Протокол SSH, як і решта протоколів віддаленого управління, побудований із використанням клієнт-серверного підходу. SSH-клієнти та SSH-сервери доступні для більшості мережних операційних систем.

Протокол SSH належить до прикладного рівня моделі OSI та прикладного рівня стеку TCP/IP. Для організації інформаційного обміну SSH-сервер використовує порт 22 TCP. Існує дві версії протоколу: SSH-1 (1995 р.) та SSH-2 (1996 р.). Версія SSH-2 є більш безпечною у порівнянні з SSH-1, тому набула більшого поширення. На сьогодні, коли йде мова про протокол SSH, то мається на увазі саме SSH-2.

Як стандарт мереж TCP/IP протокол SSH був затверджений IETF у 2000 р. Спочатку SSH був описаний у таких стандартах: RFC-4250 „The Secure Shell (SSH) Protocol Assigned Numbers", RFC-4251 „The Secure Shell (SSH) Protocol Architecture", RFC-4252 „The Secure Shell (SSH) Authentication Protocol", RFC-4253 „The Secure Shell (SSH) Transport Layer Protocol", RFC-4254 „The Secure Shell (SSH) Connection Protocol", RFC-4255 „Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints", RFC-4256 „Generic Message Exchange Authentication for the Secure Shell Protocol (SSH)", RFC-4335 „The Secure Shell (SSH) Session Channel Break Extension", RFC-4344 „The Secure Shell (SSH) Transport Layer Encryption Modes", RFC-4345 „Improved Arcfour Modes for the Secure Shell (SSH) Transport Layer Protocol".

На сьогодні наведений перелік доповнено стандартами, що пов'язані з криптографічним захистом: RFC-4419 „Diffie-Hellman Group Exchange for the Secure Shell (SSH) Transport Layer Protocol", RFC-4432 „RSA Key Exchange for the Secure Shell (SSH) Transport Layer Protocol", RFC-4462 „Generic Security Service Application Program Interface (GSS-API) Authentication and Key Exchange for the Secure Shell (SSH) Protocol", RFC-4716 „The Secure Shell (SSH) Public Key File Format", RFC-5656 „Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer".

Архітектурно у протоколі SSH виділяють три рівні:

– транспортний рівень (Transport Layer);

– рівень аутентифікації користувача (Authentication Layer);

– рівень з'єднання (Connection Layer).

Протокол транспортного рівня забезпечує аутентифікацію сервера, конфіденційність і цілісність даних з гарною естафетною передачею. Додатково може підтримуватися стиснення даних. Протокол аутентифікації користувача дає змогу серверу аутентифікувати клієнта. Протокол з'єднання мультиплексує шифрований тунель, створюючи в ньому кілька логічних каналів.

Для автентифікації сервера у SSH використовується протокол аутентифікації сторін на основі алгоритмів електронно-цифрового підпису RSA або DSA. Для аутентифікації клієнта також може застосовуватися ЕЦП RSA або DSA, але допускається також аутентифікація за допомогою пароля (режим зворотної сумісності з Telnet) і навіть за IP-адресою вузла (режим зворотної сумісності з rlogin). Аутентифікація за паролем залишається найпоширенішою, оскільки пароль передається зашифрованим віртуальним каналом, проте сучасна практика адміністрування віддає перевагу аутентифікації за ключами, за якої спільний секрет мережею не передається взагалі. Аутентифікація за IP-адресою небезпечна, цю можливість, як правило, відключають. Для створення загального секрету (сеансового ключа) використовується алгоритм Діффі-Хеллмана. Для шифрування переданих даних використовується симетричне шифрування, алгоритми IDEA, AES, Blowfish, DES або 3DES. Цілісність передачі даних перевіряється за допомогою CRC32 у протоколі SSH версії 1 та за допомогою HMAC-SHA1/HMAC-MD5 у протоколі SSH версії 2. У протоколі SSH також можливе застосовування функції стиснення даних, що передаються. З цією метою використовується алгоритм Лемпеля – Зіва (LZ77), який забезпечує рівень стиснення, аналогічний архіватору ZIP. Стиснення у SSH активується лише за запитом клієнта і на практиці застосовується досить рідко.

Наведений перелік алгоритмів відповідає початковим реалізаціям протоколу. У сучасних реалізаціях SSH-2 алгоритми IDEA, Blowfish, DES та 3DES не застосовуються: у OpenSSH 7.6 (2017 р.) підтримку шифрів arcfour, blowfish і CAST вилучено, а режими CBC вимкнено за замовчуванням; контроль цілісності на базі CRC32 залишився лише у протоколі SSH версії 1. Нині узгоджуються шифри ChaCha20-Poly1305, AES-GCM та AES-CTR, коди автентичності HMAC-SHA2-

256/512 (зокрема у режимі EtM), а обмін ключами здійснюється переважно за алгоритмом curve25519-sha256.

Алгоритм DSA для формування ключів нині вважається застарілим: у OpenSSH 9.8 його вимкнено за замовчуванням, а у версії 10.0 (2025 р.) вилучено остаточно. Поряд із RSA сучасні реалізації використовують ключі Ed25519 (основний тип за замовчуванням) та ECDSA.

На базі протоколу SSH розроблено і функціонує ряд інших протоколів. Зокрема, SFTP (SSH File Transfer Protocol), SCP (Secure CoPy), FISH (Files Transferred over Shell Protocol), Rsync.

Широкого використання протокол SSH набув для віддаленого доступу та адміністрування мережних пристроїв. Більшість виробників мережного обладнання (в т.ч. Cisco, Juniper, Arista, Huawei, VyOS – наступник Vyatta, та ін.) включають реалізації SSH-серверів та клієнтів у мережні операційні системи комутаторів, маршрутизаторів та інших пристроїв і саме цей протокол рекомендують використовувати.

Рекомендації щодо підвищення рівня захищеності віддалених мережних підключень на базі протоколів Telnet та SSH

Багатьма виробниками обладнання розроблені базові рекомендації, що стосуються підвищення рівня захищеності віддалених мережних підключень до комунікаційних пристроїв на базі протоколів Telnet та SSH. Часто ці рекомендації поєднуються з рекомендаціями щодо політик застосування засобів аутентифікації (парольного доступу).

Базові рекомендації щодо застосування засобів аутентифікації є такими:

1. Забезпечувати формування та дотримання політики стійких до зламу паролів.
2. Забезпечувати застосування аутентифікації з використанням механізму користувачів із зазначенням відповідного рівня привілеїв.
3. Вилучити/відключити облікові записи адміністраторів/користувачів, що створені за замовчуванням.
4. Змінити паролі, що створені за замовчуванням.
5. Надавати користувачам доступ до пристроїв/ресурсів (авторизувати користувачів) із мінімально необхідним рівнем привілеїв.
6. Забезпечити періодичну зміну паролів користувачів.

7. Обмежено застосовувати засоби протоколу SNMP (та інших подібних протоколів), у повідомленнях яких здійснюється передача паролів адміністраторів.

8. Забезпечити контроль параметрів, що встановлені на пристроях для виконання процедур відновлення паролів.

Загальні рекомендації щодо захисту віддалених підключень є такими:

1. Забезпечити, за можливості, застосування засобів віддаленого доступу та адміністрування, які базуються на захищених протоколах, зокрема: SSH, SCP, SSL, OTP тощо.

2. Забезпечити застосування засобів віддаленого доступу та адміністрування, що базуються на незахищених протоколах Telnet, Rlogin (та інших незахищених протоколах: Syslog, SNMP, FTP, TFTP тощо), лише на мережних керуючих підключеннях (OOB-підключеннях, що, як правило, забезпечують підключення лише легітимних користувачів/адміністраторів).

3. Використовувати як VLAN керування пристроєм нестандартну VLAN (будь-яку VLAN, окрім Default VLAN – VLAN 1, що створюється за замовчуванням).

4. Обмежити можливості віддалених підключень за допомогою списків доступу, зокрема у випадках, якщо застосування мережних керуючих підключень (OOB) неможливе.

Рекомендації щодо застосування засобів протоколу SSH є такими:

1. Заборонити віддалений доступ під паролем основного адміністратора (root-доступ).

2. Заборонити підключення з пустим паролем або відключення входу за паролем.

3. Встановити нестандартний порт для SSH-сервера.

4. Забезпечити використання довгих SSH версії 2 RSA-ключів (як мінімум 2048, бажано 3072 біт і більше) або ключів Ed25519.

5. Сформувати та застосувати список IP-адрес, із яких дозволений доступ.

6. Заборонити доступ із потенційно небезпечних IP-адрес.

7. Відмовитися від використання поширених або широковідомих системних імен користувачів для віддалених підключень.

8. Забезпечити регулярний перегляд та аналіз повідомлень про спроби та помилки аутентифікації.

9. Встановити та забезпечити функціонування систем виявлення і попередження втручань.

10. Забезпечити використання засобів-пасток, які підроблюють SSH-сервіс (HoneyPots).

11. Надавати перевагу аутентифікації за ключами (Ed25519 або RSA завдовжки 3072 біт і більше) та, за можливості, вимикати аутентифікацію за паролем.

12. Заборонити використання протоколу SSH версії 1 (на пристроях Cisco – командою ip ssh version 2).

Серверні та клієнтські засоби організації віддаленого доступу до мережних пристроїв із використанням протоколів Telnet та SSH

Як відомо, протокол віддаленого доступу Telnet працює з використанням клієнт-серверного підходу. У більшості сучасних ОС наявні серверні та клієнтські програмні модулі, які забезпечують роботу цього протоколу. У деяких ОС згадані модулі є невід'ємними їх частинами та активуються за замовчуванням під час початкового встановлення системи, в інших ОС – потрібне виконання певних додаткових дій щодо їх встановлення та активації. У більшості ОС існує можливість встановлення і використання Telnet-серверів та Telnet-клієнтів сторонніх виробників. Як правило, і серверні, і клієнтські додатки протоколу Telnet використовують інтерфейс командного рядка.

Під час розробки протоколу Telnet проблемам захисту інформації не приділяли достатньої уваги, тому сеанси інформаційного обміну, які організовані за даним протоколом, не є захищеними від дій зловмисників. З метою підвищення рівня інформаційної безпеки засобами даного протоколу не рекомендується користуватися у відкритих (незахищених) мережних середовищах. Як виняток, можливе використання Telnet у захищених сегментах мереж.

Основною альтернативою використанню протоколу Telnet, яка забезпечує високий рівень захисту інформаційного обміну в ході організації віддаленого доступу як у відкритих, так і у захищених мережних середовищах, сьогодні є протокол SSH. Цей протокол, як і протокол Telnet, працює з використанням клієнт-серверного підходу. На жаль, у багатьох мережних ОС відсутні вбудовані програмні модулі, які забезпечують його роботу.

На ринку програмного забезпечення наявна велика кількість як комерційних, так і вільнорозповсюджуваних SSH-серверів та SSH-клієнтів для різних ОС. Деякі з них орієнтовані на використання лише в певних ОС, деякі є кросплатформними. Найбільш поширеними

SSH-серверами є OpenSSH (зокрема вбудований OpenSSH Server в ОС Windows), Bitvise SSH Server, CopSSH, Dropbear, GoAnywhere Services, MobaSSH SSH Server, Pragma Fortress SSH Server, Tectia SSH Server, wolfSSH та Apache MINA SSHD. Найбільш поширеними SSH-клієнтами є OpenSSH, PuTTY, SecureCRT та інші.

Узагальнена інформація про вбудовані серверні та клієнтські засоби організації віддаленого доступу сучасних мережних ОС наведена у табл. 1.

Таблиця 1

Вбудовані серверні та клієнтські засоби організації віддаленого доступу

ОС	Telnet		SSH	
	Сервер	Клієнт	Сервер	Клієнт
Windows XP	+	+	–	–
Windows 10/11	–	+*	+*	+
Linux	+	+	+	+
Cisco IOS	+	+	+	+
Juniper JunOS	+	+	+	+

* необхідні додаткові дії щодо встановлення та активації компонента в системі; у Windows 10/11 клієнт SSH наявний за замовчуванням, а сервер Telnet не постачається взагалі

Встановлення та використання серверних та клієнтських додатків протоколів Telnet та SSH в ОС Windows

В ОС Windows XP сервер та клієнт протоколу Telnet встановлюються автоматично під час початкового встановлення системи. В ОС Windows 7/8 необхідно виконати додаткові дії щодо їх встановлення. У Windows 7 інсталяція Telnet-сервера та Telnet-клієнта проводиться через „Панель керування” → „Програми та компоненти” → „Увімкнення або вимкнення компонентів Windows” → „Сервер Telnet” („Клієнт Telnet”). В ОС Windows запуск (або зупинка) Telnet-сервера здійснюється через „Панель керування” → „Адміністрування” → „Служби” (запускається додаток **tlntsvr**). Альтернативним способом запуску/зупинки Telnet-сервера є використання мережних команд **net start tlntsvr** та **net stop tlntsvr** відповідно.

Слід зазначити, що наведений порядок дій стосується ОС Windows 7/8. Сервер Telnet (служба tlntsvr та додаток tlntadmn) не постачається у складі Windows 10/11 – серед додаткових компонентів цих систем наявний лише клієнт Telnet, а довідку Microsoft щодо додатка tlntadmn переведено до архіву і вона поширюється лише на

Windows Vista/7/8 та Windows Server 2008–2012. Наведені нижче приклади роботи з Telnet-сервером ОС Windows слід розглядати як довідкові; у цій лабораторній роботі Telnet-сервером виступає комунікаційний пристрій Cisco.

Починаючи з Windows 10 (складання 1809) та Windows Server 2019, засоби OpenSSH постачаються як додатковий компонент системи (Feature on Demand): клієнт SSH встановлено за замовчуванням, сервер SSH потребує встановлення та активації. У Windows Server 2025 сервер SSH уже встановлено, але не активовано.

Для адміністрування Telnet-сервера в ОС Windows передбачено використання специфічного текстового додатка **tlntadmn.exe**. Запущений без ключів додаток (рис. 2) показує налагоджені за замовчуванням параметри роботи сервера. Для зміни параметрів цей додаток може використовуватися з відповідними ключами (рис. 3).

Telnet-клієнти ОС Windows реалізуються у вигляді текстових та графічних утиліт. Серед вбудованих в ОС графічних Telnet-клієнтів можна відмітити Microsoft Telnet (ОС Windows 9x) та Hyper Terminal (усі версії Windows). У багатьох випадках адміністратори використовують багатофункціональні термінальні клієнти сторонніх виробників, які дають змогу здійснювати віддалені мережні підключення не лише за протоколом Telnet, а й за іншими протоколами (зокрема, SSH та rlogin). Часто такі клієнти забезпечують роботу і консольних підключень. Найбільш відомими термінальними клієнтами для ОС Windows є PuTTY (www.putty.org) та SecureCRT (www.vandyke.com).

До сучасних термінальних клієнтів для ОС Windows належать також вбудований клієнт OpenSSH (команда ssh) у поєднанні з додатком Windows Terminal, а також MobaXterm та Solar-PuTTY; для копіювання файлів захищеними каналами застосовуються WinSCP та команда scp.

```
C:\>tlntadmn
Параметри localhost
Клавішу Alt перепризначено на сполучення 'CTRL+A':     YES
Тайм-аут простою сеансу                          :     1 годин
Максимум підключень                              :     2
Порт Telnet                                      :     23
Максимальна кількість спроб входу в систему      :     3
Дії при відключенні                              :     YES
Режим роботи                                     :     Console
Спосіб перевірки достовірності                   :     NTLM, Password
```

```
Домен за замовчуванням                  :       WS_01
Стан                                    :       Працює
C:\>
```

Рис. 2. Налагоджені за замовчуванням параметри роботи Telnet-сервера в ОС Windows

```
C:\>tlntadmn /?
Використання: tlntadmn [ім'я комп'ютера] [загальні_параметри] start | stop | pause
| continue | -s | -k | -m | config параметри_налаштування
                                Для роботи з усіма сеансами використовуйте
                                значення 'all'.
    -s код_сеансу               Виведення відомостей про сеанс.
    -k код_сеансу               Припинення сеансу.
    -m код_сеансу               Надсилання повідомлення сеансу.
    config                      Налаштування параметрів сервера telnet.
загальні_параметри:
    -u користувач               Користувач, облікові дані якого будуть
                                використовуватися
    -p пароль                   Пароль користувача
параметри_налаштування:
    dom = домен                 Домен за замовчуванням для імен користувачів
    ctrlakeymap = yes|no        Зіставлення клавіші ALT
    timeout = гг:хх:сс          Тайм-аут для сеансу, що простоює
    timeoutactive = yes|no      Увімкнення тайм-ауту сеансу, що простоює.
    maxfail = кількість_спроб   Максимальна кількість невдалих спроб входу
                                до відключення.
    maxconn = кількість_підключень  Максимальна кількість підключень.
    port = число                Задає порт telnet.
    sec = [+/-]NTLM [+/-]passwd Задає механізм перевірки достовірності
    mode = console|stream       Задає режим роботи.
C:\>
```

Рис. 3. Ключі для налагодження параметрів роботи Telnet-сервера в ОС Windows

Запуск вбудованого в ОС Windows Telnet-клієнта проводиться за допомогою командного рядка. Для безпосереднього зазначення параметрів підключення можуть використовуватися відповідні ключі (рис. 4). Слід зазначити, що для різних версій ОС Windows можуть бути наявні відмітності у переліку та використанні ключів.

Якщо додаток запускається без ключів, то адміністратор потрапляє у командний рядок Telnet-клієнта. У цьому разі параметри мережного підключення налагоджуються за допомогою відповідних команд (рис. 5). Слід звернути уваги на особливості налагодження параметрів Telnet-з'єднання за допомогою команди **set** (рис. 6).

```
C:\>telnet /?
telnet [-a][-e Символ][-f Файл_входу][-l Ім'я][-t Тип][Вузол [Порт]]
 -l      Ім'я користувача для входу у віддалену систему за умови, що
         підтримується параметр TELNET ENVIRON.
 -a      Спроба автоматичного входу в систему. Як і ключ -l, але використовує
         поточне ім'я користувача, під яким виконано вхід у систему.
```

```
 -e      Службовий символ перемикання режиму введення у вікні telnet-клієнта.
 -f      Ім'я файлу з боку клієнта для виконання входу в систему.
 -t      Тип telnet-термінала.
         Підтримуються лише 4 типи терміналів: vt100, vt52, ansi та vtnt.
 Вузол   Ім'я вузла або IP-адреса віддаленого комп'ютера, до якого виконується
         підключення.
 Порт    Номер порту або ім'я служби.
C:\>
```

Рис. 4. Ключі для налагодження параметрів підключення Telnet-клієнта в ОС Windows 7

```
Ласкаво просимо до програми-клієнта Microsoft Telnet
Символ перемикання режиму: 'CTRL+]'
Microsoft Telnet> ?
Команди можна скорочувати. Підтримуються такі команди:
c    - close                 закрити поточне підключення
d    - display               показати параметри операції
o    - open ім'я_вузла [Порт]  підключитися до вузла (за замовчуванням Порт = 23)
q    - quit                  вийти з telnet
set  - set                   установити параметри (,,set ?,, для виведення їх списку)
sen  - send                  надіслати рядки на сервер
st   - status                вивести відомості про поточний стан
u    - unset                 скинути параметри (,,unset ?,, для виведення їх списку)
?/h  - help                  вивести довідку
Microsoft Telnet>
```

Рис. 5. Команди Telnet-клієнта ОС Windows 7

```
Microsoft Telnet> set ?
bsasdel     символ <BackSpace> надсилатиметься як символ <Delete>
crlf        режим повернення каретки; призводить до надсилання символів CR & LF
delasbs     символ <Delete> надсилатиметься як символ <BackSpace>
escape x    де x – символ перемикання в режим telnet-термінала і назад
localecho   увімкнення локального відлуння.
logfile x   де x – файл журналу поточного клієнта
logging     ведення журналу
mode x      де x – консоль або потік
ntlm        увімкнення перевірки достовірності NTLM.
term x      де x – ansi, vt100, vt52 або vtnt
Microsoft Telnet>
```

Рис. 6. Параметри команди **set** Telnet-клієнта ОС Windows 7

Встановлення та використання серверних та клієнтських додатків протоколів віддаленого доступу Telnet та SSH в ОС Linux/Unix

Більшість Linux/Unix-подібних ОС у своєму складі мають засоби забезпечення роботи протоколу віддаленого доступу Telnet. У деяких з них Telnet-сервер та Telnet-клієнт Telnet встановлюються і активуються автоматично при початковому встановленні системи. Через

проблеми безпеки протоколу у більшості сучасних ОС не виконується встановлення та активація роботи Telnet-сервера, а виконується лише встановлення Telnet-клієнта.

Для запуску Telnet-сервера використовується системна служба (демон) **telnetd**. Запуск може здійснюватися як в автоматичному, так і у ручному режимі. Додаток Telnet-клієнта встановлюється автоматично у більшості ОС Linux/Unix. Його запуск проводиться за допомогою командного рядка. За допомогою відповідних ключів існує можливість налагодити відповідні параметри підключення. Перелік ключів для запуску Telnet-клієнта ОС Linux MicroCore наведено на рис. 7.

```
tc@box:~$ telnet
BusyBox v1.19.0 (2011-08-14 21:05:38 UTC) multi-call binary.
Usage: telnet [-a] [-l USER] HOST [PORT]
Connect to telnet server
        -a      Automatic login with $USER variable
        -l USER Automatic login as USER
tc@box:~$
```

Рис. 7. Ключі для налагодження параметрів підключення Telnet-клієнта в ОС Linux MicroCore

Як безпечна альтернатива протоколу Telnet в Linux/Unix-системах широко використовується протокол віддаленого доступу SSH. На ринку програмних додатків існує досить багато відкритих розробок протоколу SSH. Найбільш поширеною і такою, що динамічно розвивається, є реалізація відома як OpenBSD Secure Shell або OpenSSH (www.openssh.org). У більшості сучасних Linux/Unix-подібних ОС SSH-сервер та SSH-клієнт встановлюються та активуються автоматично під час початкового встановлення системи. У багатьох ОС користувач має можливість керувати процесом встановлення SSH-сервера та SSH-клієнта у діалоговому режимі.

Сценарій налагодження та активації OpenSSH сервера в ОС Linux MicroCore наведений нижче.

```
root@box:#ls /mnt/hda1/tce/optional/openssh*
/mnt/sda1/tce/optional/openssh.tcz
/mnt/sda1/tce/optional/openssh.tcz.dep
/mnt/sda1/tce/optional/openssh.tcz.md5.txt
root@box:#mv /usr/local/etc/ssh/sshd_config.example usr/local/etc/ssh/sshd_config
root@box:#/usr/local/etc/init.d/openssh start
root@box:#/
```

Для автоматичної активації OpenSSH-сервера при запуску ОС Linux MicroCore необхідно відредагувати та зберегти відповідні конфігураційні файли системи. Внесення змін у зазначені файли виконується або за допомогою системної команди **echo**, або за допомогою текстового редактора. Оскільки Linux MicroCore є специфічно побудованою системою, то для збереження змін конфігурації слід скористатися спеціальною командою **filetool.sh -b**.

Сценарій виконання дій щодо внесення змін у конфігураційні файли OpenSSH-сервера в ОС Linux MicroCore наведено нижче.

```
root@box:#echo ,,openssh.tcz,, >> /mnt/hda1/tce/onboot.lst
root@box:#echo ,,/usr/local/etc/init.d/openssh start,, >> /opt/bootlocal.sh
root@box:#echo ,,/usr/local/etc/ssh,, >> /opt/.filetool.lst
root@box:#/usr/bin/filetool.sh -b
root@box:#
```

Перелік ключів для запуску SSH-клієнта ОС Linux MicroCore наведено на рис. 8.

```
root@box:~# ssh
usage: ssh [-1246AaCfgKkMNnqsTtVvXxYy] [-b bind_address] [-c cipher_spec]
           [-D [bind_address:]port] [-e escape_char] [-F configfile]
           [-I pkcs11] [-i identify_file]
           [-L [bind_address:]port:host:hostport]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-R [bind_address:]port:host:hostport] [-S ctl_path]
           [-W host:port] [-w local_tun[:remote_tun]
           [user@]hostname [command]
root@box:~#
```

Рис. 8. Ключі для налагодження параметрів підключення SSH-клієнта в ОС Linux MicroCore

Порядок налагодження сервера та клієнта протоколу Telnet на обладнанні Cisco

Налагодження функціонування Telnet-сервера на пристроях Cisco для забезпечення організації віддаленого доступу може здійснюватися з використанням трьох підходів:

– безпарольний вхід;

– вхід із використанням паролів на мережні підключення (та командні режими);

– вхід із використанням механізму користувачів.

Для безпарольного входу порядок виконання етапів налагодження є таким:

1. Обрати мережне (мережні) підключення для подальшої активації віддаленого доступу за протоколом Telnet (обов'язково).

2. Відключити використання аутентифікації для входу в систему для обраного мережного підключення/обраних мережних підключень (обов'язково).

3. Активувати можливість Telnet-підключення для відповідного мережного підключення/відповідних мережних підключень (обов'язково).

4. Налагодити додаткові параметри (тайм-аути, системні повідомлення тощо) для відповідного мережного підключення/відповідних мережних підключень (необов'язково).

Для входу з використанням паролів на мережні підключення порядок виконання етапів налагодження є таким:

1. Обрати мережне (мережні) підключення для подальшої активації віддаленого доступу за протоколом Telnet (обов'язково).

2. Створити пароль входу для відповідного мережного підключення/відповідних мережних підключень та паролі на командні режими (обов'язково).

3. Активувати використання парольної аутентифікації для відповідного мережного підключення/відповідних мережних підключень (обов'язково).

4. Активувати можливість Telnet-підключення для відповідного мережного підключення/відповідних мережних підключень (обов'язково).

5. Налагодити додаткові параметри (тайм-аути, системні повідомлення тощо) для відповідного мережного підключення/відповідних мережних підключень (необов'язково).

Для входу з використанням механізму користувачів порядок виконання етапів налагодження є таким:

1. Створити локального користувача із зазначенням відповідного рівня привілеїв та пароля (обов'язково).

2. Обрати мережне (мережні) підключення для подальшої активації віддаленого доступу за протоколом Telnet (обов'язково).

3. Активувати використання парольної аутентифікації з використанням локальної бази користувачів для відповідного мережного підключення/відповідних мережних підключень (обов'язково).

4. Активувати можливість Telnet-підключення для відповідного мережного підключення/відповідних мережних підключень (обов'язково).

5. Налагодити додаткові параметри (тайм-аути, системні повідомлення тощо) для відповідного мережного підключення/відповідних мережних підключень (необов'язково).

Для організації звичайного підключення для Telnet-клієнта не потрібно проводити налагодження параметрів. За потреби організації специфічного складного підключення існує можливість налагодження певних специфічних параметрів, наприклад, інтерфейсу виходу підключення на маршрутизаторі.

За звичайного використання для Telnet-клієнта немає необхідності виконувати налагодження параметрів підключення. За потреби можливе використання великої кількості специфічних параметрів підключення (наприклад, тип терміналу, перевірка достовірності, інтерфейс виходу для маршрутизатора). Перелік параметрів можна визначити з довідки системи. Налагодження параметрів здійснюється безпосередньо у командному рядку під час організації сеансу. Слід зазначити, що за допомогою Telnet-клієнта можна підключатися не лише до Telnet-сервера, а й до серверів та складових інших мережних протоколів стеку TCP/IP (зокрема, поштових протоколів SMTP, POP3, протоколу маршрутизації BGP і т.д.).

Порядок налагодження сервера та клієнта протоколу SSH на обладнанні Cisco

Налагодження функціонування SSH-сервера на пристроях Cisco для забезпечення віддаленого доступу може здійснюватися з використанням двох підходів:

– з використанням імені пристрою та імені домену;

– з використанням ключових пар RSA (без використання імені пристрою та імені домену).

Слід зазначити, що одним із обов'язкових попередніх етапів налагодження SSH-сервера є створення локального користувача з зазначенням відповідного рівня привілеїв та пароля.

Для підходу з використанням імені пристрою та імені домену порядок виконання етапів налагодження є таким:

1. Виконати іменування пристрою (обов'язково).

2. Виконати іменування домену (обов'язково).

3. Згенерувати SSH-ключ (ключову пару RSA), який буде використовуватися у процесі роботи (обов'язково).

4. Налагодити додаткові параметри SSH-сервера: версію протоколу, час тайм-ауту, кількість спроб аутентифікації та ін. (необов'язково).

5. Обрати мережне (мережні) підключення для подальшої активації віддаленого доступу за протоколом SSH (обов'язково).

6. Активувати використання локальної бази даних користувачів для обраного мережного підключення/обраних мережних підключень (обов'язково).

7. Активувати можливість SSH-підключення для відповідного мережного підключення / відповідних мережних підключень (обов'язково).

8. Налагодити додаткові параметри (тайм-аути, системні повідомлення тощо) для відповідного мережного підключення/відповідних мережних підключень (необов'язково).

Для підходу з використанням ключових пар RSA (без використання імені пристрою та імені домену) порядок виконання етапів налагодження є таким:

1. Створити ключову пару RSA, яка буде використовуватися у процесі роботи (обов'язково).

2. Згенерувати ключову пару RSA із зазначенням довжини ключа (обов'язково).

3. Налагодити додаткові параметри SSH-сервера: версію протоколу, час тайм-ауту, кількість спроб аутентифікації та ін. (необов'язково).

4. Обрати мережне (мережні) підключення для подальшої активації віддаленого доступу за протоколом SSH (обов'язково).

5. Активувати використання локальної бази даних користувачів для обраного мережного підключення/обраних мережних підключень (обов'язково).

6. Активувати можливість SSH-підключення для відповідного мережного підключення/відповідних мережних підключень (обов'язково).

7. Налагодити додаткові параметри (тайм-аути, системні повідомлення тощо) для відповідного мережного підключення/відповідних мережних підключень (необов'язково).

За звичайного використання для SSH-клієнта не потрібно виконувати налагодження параметрів підключення. Специфічні параметри підключення встановлюються за рахунок використання ключів у командному рядку клієнта.

Загальні команди налагодження функціонування протоколів віддаленого доступу на пристроях Cisco

Для налагодження функціонування протоколів віддаленого доступу (зокрема, Telnet та SSH) на пристроях Cisco використовуються як деякі загальні для всіх протоколів команди, так і характерні лише для певного протоколу команди. До загальних команд належать такі команди: **password**, **username**, **login**, **transport**, **rotary**, **autocommand, login**, **security authentication** та похідні від них команди.

Команди **login, password**, **username** призначені для налагодження параметрів аутентифікації для певного мережного підключення, команди групи **transport** призначені для дозволу/заборони віддалених підключень до/з пристрою з використанням різних мережних протоколів. Команда **rotary** відповідає за налагодження нестандартних портів для підключень. Команда **autocommand** дає можливість налагодити виконання певної команди режиму користувача після підключення. Для управління сеансами мережних протоколів можуть використовуватися як певні комбінації клавіш (для призупинення сесії **Ctrl+Shift+6, x**), так і певні команди (повернення до сеансу – команда **resume**, завершення сеансу – команда **disconnect**).

Важливими командами, що дають змогу здійснювати контроль та журналювання процесу віддалених підключень є команди, похідні від команд **login** та **security**: **login block-for**, **login delay login on-failure log**, **login on-success log**, **login quiet-mode** та **security authentication failure rate** відповідно. Команда **login block-for** застосовується для блокування можливості підключення до пристрою на певний період часу, якщо перевищено кількість спроб підключення на вставлений

інтервал часу. Команда **login delay** зазначає затримку між спробами підключення. Команди **login on-failure log** та **login on-success log** застосовуються для активації журналювання подій при підключенні та аутентифікації користувача у системі. Вказані команди активують журналювання подій про неуспішні та успішні спроби відповідно. Команда **login quiet-mode access-class** застосовується для активації списків доступу для віддалених підключень. Команда **security authentication failure** встановлює кількість дозволених невдалих спроб входу в систему (за хвилину), перевищення якої викличе генерацію повідомлення для журналювання подій. Призначення та синтаксис усіх вищезгаданих команд наведено нижче.

Синтаксис команди **transport input** (режим конфігурування лінії):

transport input {*value* | *values*},

де ***value*** – параметр, який може набувати значень **all**, **lapb-ta**, **lat**, **mop**, **none**, **pad**, **rlogin**, **ssh**, **telnet**, **udptn**, **v120**; залежно від версії IOS можливі й інші значення;

***values* –** рядок параметрів, що формується із значень **lapb-ta**, **lat**, **mop**, **pad**, **rlogin**, **ssh**, **telnet**, **udptn**, **v120**;

all – всі протоколи;

lapb-ta – термінальний адаптер протоколу LAPB;

lat – протокол DEC LAT;

mop – протокол DEC MOP Remote Console Protocol;

none – жоден із протоколів;

pad – протокол X.3 PAD;

rlogin – протокол Rlogin;

ssh – протокол SSH;

telnet – протокол Telnet;

udptn – асинхронний UDPTN через UDP протокол;

v120 – Асинхронне підключення через ISDN.

Синтаксис команди **transport output** (режим конфігурування лінії):

transport output {*value* | *values*}.

Параметри команди аналогічні параметрам попередньої команди.

Синтаксис команди transport preferred (режим конфігурування лінії):

transport preferred value.

Параметр команди аналогічний параметру ***value*** попередньої команди, за винятком значення **all**.

Синтаксис команди **rotary** (режим конфігурування лінії):

rotary *value* [queued [by-role [round-robin] | round-robin]] | round-robin [queued [by-role]],

де ***value*** – значення номера групи, що задається для номера порту; число з діапазону 0 … 127.

queued – параметр, який вказує на необхідність використання черги, коли група заповнена;

round-robin – параметр, який вказує на необхідність кругового вибору;

by-role – параметр, який вказує на необхідність вибору за ролями.

Синтаксис команди **autocommand** (режим конфігурування лінії):

autocommand { LINE | no-suppress-linenumber LINE },

де **LINE** – текстовий рядок, який містить команду (режиму EXEC), що буде автоматично виконуватися;

no-suppress-linenumber – параметр, який призначений для активації виведення повідомлення.

Синтаксис команди **login block-for** (режим глобального конфігурування):

login block-for *block_value* attempts *attempts_value* within *interval_value*,

де ***block_value*** – значення інтервалу часу (с), на який буде заблоковано можливість виконання підключення; може змінюватися у межах від 1 до 65535 с;

attempts – службова конструкція, за допомогою якої зазначається максимально можлива кількість спроб підключення на заданий інтервал часу;

attempts_value – максимальне значення кількості спроб підключення на встановлений інтервал часу; може змінюватися у межах від 1 до 65535;

within – службова конструкція, за допомогою якої зазначається встановлений для можливої кількості спроб підключення інтервал часу;

interval_value – значення інтервалу часу (с), на якому задається максимальне значення кількості спроб підключення; може змінюватися у межах від 1 до 65535 с.

Синтаксис команди **login delay** (режим глобального конфігурування):

login delay *delay_value*,

де ***delay_value*** – значення інтервалу затримки (с) між спробами підключення; може змінюватися у межах від 1 до 10 с; за замовчуванням становить 1 с.

Синтаксис команди **login on-failure log** (режим глобального конфігурування):

login on-failure log [every *login_value*],

де **every** – службова конструкція, за допомогою якої зазначається необхідність виконання журналювання неуспішних спроб входу в систему;

login_value – числове значення кількості спроб підключення, може змінюватися у межах від 1 до 65535.

Синтаксис команди **login on-success log** (режим глобального конфігурування):

login on-success log [every *login_number*],

де **every** – службова конструкція, за допомогою якої зазначається необхідність виконання журналювання успішних спроб входу в систему;

login_value – числове значення кількості спроб підключення, може змінюватися у межах від 1 до 65535.

Синтаксис команди **login quiet-mode access-class** (режим глобального конфігурування):

login quiet-mode access-class { *acl_name* | *acl_number* },

де ***acl_name*** – текстова назва списку доступу;

acl_number – номер списку доступу.

Синтаксис команди **security authentication failure rate** (режим глобального конфігурування):

security authentication failure rate *rate_value* log,

де ***rate_value*** – кількість дозволених невдалих спроб входу в систему (аутентифікації) за хвилину, перевищення якої викличе генерацію повідомлення для журналювання подій;

log – службова конструкція, за допомогою якої активується можливість виконання журналювання подій, пов’язаних із невдалими спробами входу в систему.

Команди налагодження функціонування протоколу Telnet на пристроях Cisco

Для налагодження параметрів функціонування протоколу Telnet на пристроях Cisco використовуються спеціалізовані команди **ip telnet**: **ip telnet comport**, **ip telnet hidden**, **ip telnet quiet**, **ip telnet source-interface**, **ip telnet timeout retransmit**, **ip telnet tos**. Команди групи **ip telnet comport** призначені для налагодження параметрів функціонування засобів протоколу Telnet згідно із RFC-2217. Команда **ip telnet hidden** призначена для відключення виведення IP-адрес або назв вузлів протоколу Telnet. Команда **ip telnet quiet** відповідає за вимкнення виведення неінформативних повідомлень. Команда **ip telnet source-interface** призначена для встановлення інтерфейсу виходу сеансів протоколу Telnet. Згадана команда застосовується, як правило, на маршрутизаторах та комутаторах третього рівня. Команда **ip telnet timeout retransmit** відповідає за встановлення значення тайм-ауту повторної передачі. Команда **ip telnet tos** застосовується для встановлення значення типу сервісу (TOS, Type Of Service). Призначення та синтаксис усіх вищезгаданих команд наведено нижче.

Синтаксис команди **ip telnet comport** (режим глобального конфігурування):

ip telnet comport { disconnect delay *disconnect_value* | enable | flow level *flow_value* | receive window *window_value* },

де ***disconnect_value*** – значення інтервалу затримки перед закриттям TCP-з'єднання (с), число з діапазону 0 … 360;

flow_value – кількість символів для буферизації на пристрої перед відправленням повідомлення RFC-2217 SUSPEND; число з діапазону 0 … 1023;

window_value – максимальне значення вікна отримання TCP; число з діапазону 1 … 4128.

Синтаксис команди ip telnet hidden (режим глобального конфігурування):

ip telnet hidden {addresses | hostnames},

де **addresses** – службова конструкція, за допомогою якої відключається виведення адрес;

hostnames – службова конструкція, за допомогою якої відключається виведення назв вузлів.

Синтаксис команди **ip telnet quiet** (режим глобального конфігурування):

ip telnet quiet.

Команда не має параметрів.

Синтаксис команди **ip telnet source-interface** (режим глобального конфігурування):

ip telnet source-interface ***interface-type interface-id***,

де ***interface-type*** – тип інтерфейсу, може набувати значень **Ethernet**, **FastEthernet**, **Gigabit Ethernet**, **Serial**, **Loopback**, **Tunnel**, **Vlan** та ін.;

interface-id – ідентифікатор інтерфейсу, може мати одночислове позначення ***number*** (номер інтерфейсу), двочислове позначення ***module/number*** (номер модуля (адаптера)/номер інтерфейсу), тричислове позначення ***slot/module/number*** (номер слоту/номер модуля (адаптера)/номер інтерфейсу);

Синтаксис команди **ip telnet timeout retransmit** (режим глобального конфігурування):

ip telnet timeout retransmit ***retransmit-value***,

де ***retransmit-value*** – значення інтервалу повторної передачі (с), число з діапазону 1 … 2147483.

Синтаксис команди **ip telnet tos** (режим глобального конфігурування):

ip telnet tos ***tos-value***,

де ***tos-value*** – значення параметра TOS, число з діапазону 0h …FFh

Команди налагодження функціонування протоколу SSH на пристроях Cisco

Для налагодження параметрів функціонування протоколу SSH на пристроях Cisco використовуються команди **ip ssh authentication-retries**, **ip ssh break-string**, **ip ssh dh min size**, **ip ssh dscp, ip ssh logging events**, **ip ssh maxstartups**, **ip ssh port**, **ip ssh precedence**, **ip ssh rsa keypair-name**, **ip ssh source-interface**, **ip ssh time-out**, **ip ssh version**, **crypto key generate**, **crypto key generate rsa general-keys modulus**, **crypto key generate rsa usage-keys label** та деякі інші. Призначення та синтаксис усіх вищезгаданих команд наведено нижче.

Команда **ip ssh authentication-retries** призначена для встановлення кількості спроб аутентифікації, після якої SSH-клієнтові забороняється доступ. Команда **ip ssh break-string** відповідає за активацію обробки та встановлення зазначення текстового рядка, що передається SSH-клієнтом SSH-серверу, який надалі передає сигнал зупинки для асинхронного підключення. Команда **ip ssh logging events** призначена для активації журналювання подій протоколу SSH. Команда **ip ssh maxstartups** використовується для обмеження кількості сесій протоколу.

Команда **ip ssh port** відповідає за активацію безпечного доступу до пристрою через асинхронні підключення та встановлення початкового номера TCP-порту, що буде використовуватися для цих підключень. Команди **ip ssh dscp** та **ip ssh precedence** застосовуються для встановлення значень полів DSCP та поля Precedence IP-пакета, що переносить повідомлення протоколу SSH.

Команда **ip ssh source-interface** використовується для зазначення певного вихідного інтерфейсу для всіх SSH-сесій. Згадана команда застосовується, як правило, на маршрутизаторах та комутаторах третього рівня. Команда **ip ssh time-out** використовується для обмеження часу відповіді SSH-клієнта **(**SSH-сервер перериває з'єднання, якщо дані не передаються протягом часу очікування). Команда **ip ssh version** призначена для вказування версії протоколу SSH, що буде використовуватися у процесі роботи. За замовчуванням на пристроях Cisco активовано режим сумісності (версія 1.99), за якого пристрій приймає підключення як за протоколом SSH версії 1, так і за протоколом SSH версії 2; команда ip ssh version 2 обмежує роботу лише другою версією (див. рис. 18). Відміна дії більшості команд **ip ssh** виконується формою **no**.

Команда **ip ssh rsa keypair-name** застосовується для зазначення назви ключа RSA, що буде використовуватися у процесі роботи протоколу SSH. Ключ може формуватися двома способами: або з використанням назви пристрою та назви домену або без їх використання. Також для робіт із ключами застосовується команда **ip ssh dh min**

size, за допомогою якої задається номер групи Діффі-Хеллмана для обміну ключами.

Для роботи з ключами використовуються команди групи **crypto key**. Для генерації ключів застосовуються команди **crypto key generate**, **crypto key generate rsa general-keys modulus**, **crypto key generate rsa usage-keys label**. Для видалення ключів призначені команди **crypto key zeroize**, **crypto key zeroize rsa**.

Слід звернути увагу, що однакового результату у процесі налагодження функціонування протоколу SSH на пристроях Cisco можна досягнути у разі використання різних команд. Детальний опис дії команд можна знайти в документації виробника.

Синтаксис команди **ip ssh authentication-retries** (режим глобального конфігурування):

ip ssh authentication-retries *retries-value*,

де ***retries-value*** – значення максимальної кількості спроб аутентифікації підряд, число з діапазону 0…5; за замовчуванням встановлюється 3 спроби.

Синтаксис команди **ip ssh break-string** (режим глобального конфігурування):

ip ssh break-string *text-string*,

де ***text-string*** – текстовий рядок сигналу зупинки, за замовчуванням не встановлено.

Синтаксис команди **ip ssh dh min size** (режим глобального конфігурування):

ip ssh dh min size *dh_value*,

де ***dh_value*** – значення номер групи Діффі-Хеллмана, може набувати значень 1024 (група 1), 2048 (група 14), 4096 (група 16); за замовчуванням дорівнює 1024.

Слід зазначити, що нині мінімально допустимою вважається довжина 2048 бітів, тому значення 1024 (група 1) застосовувати не рекомендується.

Синтаксис команди **ip ssh dscp** (режим глобального конфігурування):

ip ssh dscp *dscp_value*,

де ***dscp_value*** – значення поля DSCP IP-пакета, може набувати значень від 0 до 63; за замовчуванням дорівнює 0.

Синтаксис команди **ip ssh logging events** (режим глобального конфігурування):

ip ssh logging events.

Команда не має параметрів.

Синтаксис команди **ip ssh maxstartups** (режим глобального конфігурування):

ssh ip ssh maxstartups [*max-value*],

де ***max-value*** – значення кількості сесій протоколу, число з діапазону 0…128; за замовчуванням встановлюється 128 сесій.

Синтаксис команди **ip ssh port** (режим глобального конфігурування):

ip ssh port *port-number* rotary *group*,

де ***port-number* –** значення номера порту; число з діапазону 2000 … 10000;

rotary – службова конструкція, призначена для активації доступу з використанням значення ***group***;

group – значення номера групи; число з діапазону 1 ... 127.

Синтаксис команди **ip ssh precedence** (режим глобального конфігурування):

ip ssh precedence *precedence_value*,

де ***precedence_value* –** значення поля Precedence, може набувати значень від 0 до 7; за замовчуванням дорівнює 0.

Синтаксис команди **ip ssh rsa keypair-name** (режим глобального конфігурування):

ip ssh rsa keypair-name *keypair-name-string*,

де ***keypair-name-string*** – текстовий рядок, який містить назву ключа RSA.

Синтаксис команди **ip ssh source-interface** (режим глобального конфігурування):

ip ssh source-interface *interface-type interface-id*,

де ***interface-type*** – тип інтерфейсу, може набувати значень **Serial**, **Ethernet**, **FastEthernet**, **Gigabit Ethernet**, **Serial**, **Port-channel**, **Tunnel** та ін.;

interface-id* –** ідентифікатор інтерфейсу, може мати одночислове позначення ***number (номер інтерфейсу), двочислове позначення

module/number (номер модуля (адаптера)/номер інтерфейсу), тричислове позначення ***slot/module/number*** (номер слоту/номер модуля (адаптера)/номер інтерфейсу).

Синтаксис команди **ip ssh time-out** (режим глобального конфігурування):

ip ssh time-out *seconds*,

де ***seconds*** – значення інтервалу часу очікування відповіді клієнта (с), число з діапазону 1 ... 120; за замовчуванням встановлюється 120 с.

Синтаксис команди **ip ssh version** (режим глобального конфігурування):

ip ssh version *version-number*,

де ***version-number*** – номер версії протоколу, може набувати значень 1 або 2; якщо значення не встановлене, то функціонування протоколу здійснюється у змішаному режимі.

Синтаксис команди **crypto key generate** (режим глобального конфігурування):

crypto key generate.

Команда не має параметрів.

Синтаксис команди **crypto key generate rsa general-keys modulus** (режим глобального конфігурування):

crypto key generate rsa general-keys modulus *modulus-value*,

де ***modulus-value*** – значення довжини ключа (бітів), число з діапазону 360 ... 2048; за замовчуванням генеруються ключі довжиною 512 бітів.

Синтаксис команди **crypto key generate rsa usage-keys label** (режим глобального конфігурування):

crypto key generate rsa usage-keys label *keypair-name-string* modulus *modulus-value*,

де ***keypair-name-string*** – текстовий рядок, який містить назву ключової пари RSA;

modulus-value – значення довжини ключа (бітів), число з діапазону 360 ... 2048; за замовчуванням генеруються ключі довжиною 512 бітів.

Синтаксис команди **crypto key zeroize** (режим глобального конфігурування):

crypto key zeroize (команда не має параметрів)

Синтаксис команди **crypto key zeroize rsa** (режим глобального конфігурування):

crypto key zeroize rsa *keypair-name-string*,

де ***keypair-name-string*** – текстовий рядок, який містить назву ключової пари RSA.

Основні команди моніторингу та діагностики функціонування протоколів Telnet та SSH на пристроях Cisco

Для перегляду параметрів налагоджень мережних підключень, параметрів роботи протоколів віддаленого доступу та інших параметрів використовуються різні варіанти команд **show**. Перелік основних команд та їх призначення наведені у табл. 2. Для відстеження подій та повідомлень, які генеруються протоколом SSH використовуються команди **debug ip ssh** та **debug ip ssh client**.

Таблиця 2

Перелік команд show, необхідних для діагностики параметрів мережних підключень та параметрів протоколів віддаленого доступу на пристроях Cisco

Команда	Призначення
show line	Виведення інформації про наявні підключення та їх параметри
show sessions	Виведення інформації про параметри вихідних сеансів
show users	Виведення інформації про параметри вхідних підключень (зокрема, мережних) та підключених користувачів
show ssh	Виведення інформації про параметри сеансів протоколу SSH
show ip ssh	Виведення інформації про налагоджені параметри протоколу SSH (версія, час тайм-ауту, кількість спроб аутентифікації тощо)
show login	Виведення узагальненої інформації про налагодження входу в систему
show crypto key mypubkey rsa	Виведення публічного ключа RSA
show crypto key pubkey-chain rsa	Виведення публічного ключа RSA з'єднання, що збережений на комп'ютері.
show crypto key storage	Виведення інформації про місце розміщення ключової пари
show control-plane host open-ports	Виведення інформації про відкриті TCP- та UDP-порти пристрою
show tcp	Виведення детальної інформації про встановлені з'єднання транспортного протоколу TCP (використовується для визначення IP-адрес з'єднань та номерів портів протоколів віддаленого доступу)

Модельний приклад налагодження віддаленого доступу до пристроїв Cisco з використанням протоколу Telnet

Розглянемо специфіку налагодження роботи протоколу віддаленого доступу Telnet для комунікаційних пристроїв мережі, схема якої наведена на рис. 9. У даному випадку підключення здійснюється з робочої станції WS-Control до маршрутизатора R-1 та комутатора SW-1.



Рис. 9. Приклад мережі

Під час побудови даної мережі для з'єднання пристроїв використано дані табл. 3. Для налагодження параметрів адресації пристроїв використано дані табл. 4.

Таблиця 3

Параметри інтерфейсів пристроїв для прикладу

Пристрій	Інтерфейс	Підключення до пристрою	Підключення до інтерфейсу
Маршрутизатор R-1	Fa0/0	Комутатор SW-1	Fa0/1
	Fa0/1	WAN	WAN Interface
Комутатор SW-1	Fa0/1	Маршрутизатор R-1	Fa0/0
	Fa0/2	Сервер Serv_A_1	Fa0
WS-Control	Fa0	Комутатор SW-1	Fa0/2

Таблиця 4

Параметри адресації мережі

Підмережа/ Пристрій	Інтерфейс/Мережний адаптер/Шлюз	IP-адреса	Маска підмережі	Префікс
Підмережа A	–	195.10.1.0	255.255.255.0	/24
WAN	–	196.10.1.0	255.255.255.252	/30
Маршрутизатор R-1	Інтерфейс Fa0/0	195.10.1.254	255.255.255.0	/24
	Інтерфейс Fa0/1	196.10.1.2	255.255.255.252	/30
Комутатор SW-1	Інтерфейс Vlan 1	195.10.1.250	255.255.255.0	/24
	Шлюз за замовчуванням	195.10.1.254	–	–
Робоча станція WS-Control	Мережний адаптер	195.10.1.1	255.255.255.0	/24
	Шлюз за замовчуванням	195.10.1.254	–	–

Для налагодження параметрів комунікаційних пристроїв із метою забезпечення підключення по протоколами Telnet/SSH використано дані табл. 5.

Таблиця 5

Параметри налагодження комунікаційних пристроїв

Параметр	Значення
Системний час	Поточний
Часовий пояс	Східноєвропейський
Перехід на літній час	Україна
Банер	Connection to router R-1
Кількість підключень VTY	5 (0 … 4)
Інтервал перед виведенням попередження про вихід із системи, с	30
Загальна тривалість сеансу, хв	10
Синхронне виведення журнальних повідомлень на екран	Активоване

Сценарій налагодження часових параметрів, повідомлення попередження та параметрів адресації інтерфейсів для маршрутизатора мережі R-1 наведений нижче. Сценарій налагодження параметрів комутатора мережі SW-1 подібний до сценарію для маршрутизатора R-1.

```
…
R-1>enable
R-1#clock set 10:04:00 11 dec 2022
R-1#configure terminal
R-1(config)#clock timezone EET 2
R-1(config)#clock summer-time EET recurring last monday
    october 3:00 last monday march 3:00
R-1(config)#banner motd # Connection to router R-1
    For legal users only #
R-1(config)#interface FastEthernet 0/0
R-1(config-if)#description LINK-TO-LAN-A
R-1(config-if)#ip address 195.10.1.254 255.255.255.0
R-1(config-if)#no shutdown
R-1(config-if)#exit
R-1(config)#interface FastEthernet 0/1
R-1(config-if)#description LINK_TO_WAN
R-1(config-if)#ip address 196.10.1.2 255.255.255.252
R-1(config-if)#no shutdown
```

R-1(config-if)#exit
R-1(config)#exit
R-1#
…

Сценарій налагодження віддаленого підключення за протоколом Telnet із входом без пароля (без аутентифікації) для маршрутизатора мережі R-1 наведений нижче. Слід зазначити, що у даному сценарії передбачено прямий перехід у привілейований режим за рахунок встановлення найвищого рівня привілеїв. Сценарій налагодження параметрів комутатора мережі SW-1 подібний до сценарію для маршрутизатора R-1. У практиці експлуатації мереж такий сценарій має обмежене застосування, його рекомендовано застосовувати лише у випадку, коли мережна інфраструктура є надійно захищеною. Якщо існує ймовірність перехоплення інформації під час підключення, то розглянутий сценарій застосовувати не рекомендується.

…
R-1>enable
R-1#configure terminal
R-1(config)#line vty 0 4
R-1(config-line)#no login
R-1(config-line)#transport input telnet
R-1(config-line)#privilege level 15
R-1(config-line)#logout-warning 30
R-1(config-line)#absolute-timeout 10
R-1(config-line)#logging synchronous
R-1(config-line)#exit
R-1(config)#exit
R-1#
…

Сценарій підключення/відключення до маршрутизатора R-1 з робочої станції ОС Windows за допомогою вбудованого термінального додатка Telnet наведено нижче.

```
…
C:\>telnet
Ласкаво просимо до програми-клієнта Microsoft Telnet
Символ перемикання режиму: 'CTRL+]'
Microsoft Telnet>open 195.10.1.254
Підключення до 195.10.1.254...
```

```
Connection to router R-1
For legal users only
R-1#...
…
R-1#exit
Підключення до вузла втрачено
Натисніть будь-яку клавішу ...
Microsoft Telnet>quit
C:\>
…
```

Сценарій налагодження віддаленого підключення за протоколом Telnet до маршрутизатора Cisco з використанням засобів локальної аутентифікації на базі механізму паролів на вхід до відповідних командних режимів пристрою наведений нижче. У даному сценарії застосовані паролі типу 7.

…
R-1>enable
R-1#configure terminal
R-1(config)#service password-encryption
R-1(config)#enable secret adminpass2
R-1(config)#line vty 0 4
R-1(config-line)#password adminpass1
R-1(config-line)#login
R-1(config-line)#transport input telnet
R-1(config-line)#logout-warning 30
R-1(config-line)#absolute-timeout 10
R-1(config-line)#logging synchronous
R-1(config-line)#exit
R-1(config)#exit
R-1#
…

Сценарій підключення/відключення до маршрутизатора R-1 із робочої станції ОС Windows за допомогою вбудованого термінального додатка Telnet наведено нижче. Слід зазначити, що пароль під час введення не відображається.

```
…
C:\>telnet 195.10.1.254
User Access Verification
Password:
Connection to router R-1
```

```
For legal users only
R-1>enable
Password:
R-1#...
…
R-1#exit
Підключення до вузла втрачено.
C:\>
…
```

Сценарій налагодження віддаленого підключення за протоколом Telnet до маршрутизатора Cisco з використанням засобів локальної аутентифікації на базі механізму користувачів наведений нижче. У даному сценарії застосовані паролі типу 5.

```
…
R-1>enable
R-1#configure terminal
R-1(config)#username adminer privilege 15 secret adminerpass
R-1(config)#username technic privilege 1 secret technicpass
R-1(config)#enable secret adminerpass2
R-1(config)#line vty 0 4
R-1(config-line)#login local
R-1(config-line)#transport input telnet
R-1(config-line)#logout-warning 30
R-1(config-line)#absolute-timeout 10
R-1(config-line)#logging synchronous
R-1(config-line)#exit
R-1(config)#
…
```

Сценарій підключення/відключення до маршрутизатора R-1 з робочої станції Windows за допомогою вбудованого термінального додатка Telnet наведено нижче. Слід зазначити, що пароль при введенні не відображається.

```
…
C:\>telnet 195.10.1.254
User Access Verification
Username:adminer
Password:
Connection to router R-1
For legal users only
R-1#...
…
```

```
R-1#exit
Підключення до вузла втрачено.
C:\>
…
```

Результати виконання команд моніторингу та діагностики роботи протоколу віддаленого доступу Telnet для розглянутого прикладу

З метою перегляду інформації про роботу мережних підключень, параметрів роботи протоколів віддаленого доступу та інших параметрів використовуються як загальні, так і специфічні для певного протоколу команди. Для розглянутого прикладу використано загальні команди **show line**, **show users**, **show tcp**. Результати роботи цих команд для маршрутизатора R-1 наведено відповідно на рис. 10–12. Для перевірки підключення з боку робочої станції Windows використано команду **netstat -n**. Результати роботи цієї команди наведено на рис. 13.

```
R-1#show line
   Tty Typ     Tx/Rx    A Modem  Roty AccO AccI   Uses   Noise  Overruns   Int
*     0 CTY              -    -      -    -    -      0       0     0/0       -
      1 AUX   9600/9600  -    -      -    -    -      0       0     0/0       -
*     2 VTY              -    -      -    -    -      2       0     0/0       -
      3 VTY              -    -      -    -    -      0       0     0/0       -
      4 VTY              -    -      -    -    -      0       0     0/0       -
      5 VTY              -    -      -    -    -      0       0     0/0       -
      6 VTY              -    -      -    -    -      0       0     0/0       -
R-1#
```

Рис. 10. Результат роботи команди **show line** для маршрутизатора R-1

```
R-1#show users
    Line       User       Host(s)              Idle       Location
*  0 con 0                 idle                 00:00:00
   2 vty 0     adminer    idle                 00:00:03 195.10.1.1
  Interface    User               Mode         Idle     Peer Address
R-1#
```

Рис. 11. Результат роботи команди **show users** для маршрутизатора R-1

```
R-1#show tcp
tty2, virtual tty from host 195.10.1.1
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Minimum incoming TTL 0, Outgoing TTL 255
Local host: 195.10.1.254, Local port: 23
Foreign host: 195.10.1.1, Foreign port: 1030
...
R-1#
```

Рис. 12. Результат роботи команди **show tcp** для маршрутизатора R-1

```
C:\>netstat -n
Активні підключення
```

```
  Ім’я     Локальна адреса     Зовнішня адреса     Стан
  TCP      195.10.1.1:1030     195.10.1.254:23     ESTABLISHED
C:\>
```

Рис. 13. Результат роботи команди **netstat -n** для робочої станції WS-Control

Модельний приклад налагодження віддаленого доступу до пристроїв Cisco з використанням протоколу SSH

Розглянемо специфіку налагодження роботи протоколу віддаленого доступу SSH для комунікаційних пристроїв мережі, схема якої наведена на рис. 9. У даному випадку підключення здійснюється з робочої станції WS-Control до маршрутизатора R-1 та комутатора SW-1. Під час побудови даної мережі для з’єднання пристроїв використано дані табл. 3. Для налагодження параметрів адресації пристроїв використано дані табл. 4. Для налагодження параметрів комунікаційних пристроїв з метою забезпечення підключення за протоколом SSH використано дані табл. 5.

Сценарій налагодження часових параметрів, повідомлення попередження та параметрів адресації інтерфейсів для маршрутизатора мережі R-1 аналогічний сценарію попереднього модельного прикладу. Сценарій налагодження параметрів комутатора мережі SW-1 подібний до сценарію для маршрутизатора R-1.

Сценарій налагодження віддаленого підключення за протоколом SSH до маршрутизатора Cisco з використанням імені пристрою та імені домену та з використанням засобів локальної аутентифікації на базі механізму користувачів наведений нижче. У цьому сценарії застосовані паролі типу 5.

```
…
R-1>enable
R-1#configure terminal
R-1(config)#username adminer privilege 15 secret adminerpass
R-1(config)#username technic privilege 1 secret technicpass
R-1(config)#enable secret adminerpass2
R-1(config)#ip domain-name mynet.net
R-1(config)#crypto key generate rsa general-keys modulus 1024
R-1(config)#ip ssh version 2
R-1(config)#line vty 0 4
R-1(config-line)#login local
```

R-1(config-line)#transport input ssh
R-1(config-line)#transport output ssh
R-1(config-line)#logout-warning 30
R-1(config-line)#absolute-timeout 10
R-1(config-line)#logging synchronous
R-1(config-line)#exit
R-1(config)#exit
R-1#exit
R-1>
…

Сценарій налагодження елементів захисту від атак на пристрій та налагодження підсистеми журналювання подій, пов'язаних із вдалими та невдалими спробами SSH-підключень.

…
R-1>enable
R-1#configure terminal
R-1(config)#login block-for 300 attempts 3 within 3
R-1(config)#login delay 5
R-1(config)#login on-failure log
R-1(config)#login on-success log
R-1(config)#ip ssh time-out 60
R-1(config)#ip ssh authentication-retries 5
R-1(config)#ip ssh maxstartups 5
R-1(config)#ip ssh logging events
R-1(config)#exit
R-1#exit
R-1>
…

Для підключення до маршрутизатора R-1 з робочої станції ОС Windows використано термінальний додаток PuTTY (SuperPuTTY). Сценарій підключення/відключення за допомогою цього додатка наведений нижче.

```
login as: adminer
```

```
Using keyboard-interactive authentication.
Password:
Connection to router R-1
For legal users only
R-1#...
...
R-1#exit
...
```

Сценарій налагодження віддаленого підключення за протоколом SSH до маршрутизатора Cisco з використанням ключових пар RSA та з використанням засобів локальної аутентифікації на базі механізму користувачів наведений нижче.

…
R-1>enable
R-1#configure terminal
R-1(config)#username adminer privilege 15 secret adminerpass
R-1(config)#username technic privilege 1 secret technicpass
R-1(config)#enable secret adminerpass2
R-1(config)#ip ssh rsa keypair-name MySSHkeys
R-1(config)#crypto key generate rsa usage-keys label MySSHkeys modulus 1024
R-1(config)#ip ssh version 2
R-1(config)#line vty 0 4
R-1(config-line)#login local
R-1(config-line)#transport input ssh
R-1(config-line)#transport output ssh
R-1(config-line)#exit
R-1(config)#exit
R-1#
…

У Cisco IOS існує можливість виконання підключення з одного пристрою до іншого. Для цього застосовуються вбудовані Cisco IOS Telnet та SSH-клієнти. Приклад сценарію такого підключення з комутатора SW-1 до маршрутизатора R-1 наведений нижче.

…
SW-1#ssh -v 2 -l adminer 195.10.1.1
Password:

Connection to router R-1
For legal users only
R-1#
…

Слід зазначити, що для виконання подібного сценарію на пристрої повинна бути активована можливість виконання віддалених підключень до інших пристроїв. Ця можливість активується командою **transport output**, у якій зазначається відповідний мережний протокол.

Результати виконання команд моніторингу та діагностики роботи протоколу віддаленого доступу SSH для розглянутого прикладу

З метою перегляду інформації про роботу мережних підключень, параметрів роботи протоколів віддаленого доступу та інших параметрів використовуються як загальні, так і специфічні для певного протоколу команди. Для розглянутого прикладу використано загальні команди **show line**, **show users**, **show tcp** та специфічні команди протоколу SSH **show ssh**, **show ip ssh**, **show crypto key mypubkey rsa**. Результати роботи цих команд для маршрутизатора R-1 (за умови використання імені пристрою та імені домену для підключення) наведено відповідно на рис. 14–19. Для перевірки підключення з боку робочої станції Windows використано команду **netstat -n**. Результати роботи цієї команди наведено на рис. 20.

```
R-1#show line
   Tty Typ     Tx/Rx    A Modem  Roty AccO AccI   Uses   Noise  Overruns   Int
*     0 CTY              -    -      -    -    -      0       0     0/0       -
      1 AUX   9600/9600  -    -      -    -    -      0       0     0/0       -
*     2 VTY              -    -      -    -    -      2       0     0/0       -
      3 VTY              -    -      -    -    -      0       0     0/0       -
      4 VTY              -    -      -    -    -      0       0     0/0       -
      5 VTY              -    -      -    -    -      0       0     0/0       -
      6 VTY              -    -      -    -    -      0       0     0/0       -
R-1#
```

Рис. 14. Результат роботи команди **show line** для маршрутизатора R-1

```
R-1#show users
    Line       User       Host(s)              Idle       Location
*  0 con 0                 idle                 00:00:00
```

```
   2 vty 0     adminer    idle                 00:02:57 195.10.1.1
  Interface    User               Mode         Idle     Peer Address
R-1#
```

Рис. 15. Результат роботи команди **show users** для маршрутизатора R-1

```
R-1#show tcp
tty2, virtual tty from host 195.10.1.1
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Minimum incoming TTL 0, Outgoing TTL 255
Local host: 195.10.1.254, Local port: 22
Foreign host: 195.10.1.1, Foreign port: 1030
...

R-1#
```

Рис. 16. Результат роботи команди **show tcp** для маршрутизатора R_2

```
R-1#show ssh
Connection Version Mode Encryption  Hmac         State                Username
0          2.0     IN   aes256-cbc  hmac-sha1    Session started      adminer
0          2.0     OUT  aes256-cbc  hmac-sha1    Session started      adminer
%No SSHv1 server connections running.
R-1#
```

Рис. 17. Результат роботи команди **show ssh** для маршрутизатора R-1

```
R-1#show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 60 secs; Authentication retries: 5
Minimum expected Diffie Hellman key size : 1024 bits
R-1#
```

Рис. 18. Результат роботи команди **show ip ssh** для маршрутизатора R-1

```
R-1#show crypto key mypubkey rsa
% Key pair was generated at: 22:49:37 UTC Jan 18 2017
Key name: R-1.mynet.net
 Storage Device: not specified
 Usage: General Purpose Key
 Key is not exportable.
 Key Data:
  30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00B1BDCC
  97462A4E F581491B A8F0A289 8E2EBA0C 247844A8 B53BCF0A 94700F5D 496E9E71
  541451BD C2BFFD6F 10FEFC02 5A18712C 7755EF54 1816AD97 F11F4694 8EE277A0
  FFA1F692 3D140EC6 813433A7 22C9A27C 72F3911F 6904313B 5919AA43 F086EA34
  1F4625A6 C50CC5AF CA072709 E94E2DFF 884564C2 00D43780 6D43475C 45020301 0001
% Key pair was generated at: 22:49:38 UTC Jan 18 2017
Key name: R-1.mynet.net.server
Temporary key
 Usage: Encryption Key
 Key is not exportable.
 Key Data:
  307C300D 06092A86 4886F70D 01010105 00036B00 30680261 0091A903 3A23444A
  0A4DAB64 6C1C3250 2661F503 328A9A3B 54B618E2 8FF0CF0B 6461FF1E CEDD0DB6
  4BC54A5B BD97A159 B6C82E7E E5A560C9 CCCFDB40 B7F8F898 EE3CC2E5 F1B2B716
  741B9063 DE7EB172 E2C59F81 F851A229 99450322 D2224140 91020301 0001
R-1#
```

Рис. 19. Результат роботи команди **show crypto key mypubkey rsa** для маршрутизатора R-1

```
C:\>netstat -n
Активні підключення
  Ім'я     Локальна адреса     Зовнішня адреса     Стан
  TCP      195.10.1.1:1030     195.10.1.254:22     ESTABLISHED
C:\>
```

Рис. 20. Результат роботи команди **netstat -n** для робочої станції WS-Control

Завдання на лабораторну роботу

Особливості виконання роботи у середовищі Packet Tracer

Перехоплення повідомлень (п. 8) виконується за допомогою пристрою Sniffer із групи End Devices: обраний канал зв'язку слід розірвати і з'єднати через цей пристрій (вузол – Port 0, Port 1 – порт комутатора), після чого весь трафік цього каналу відображатиметься у вікні пристрою; перелік повідомлень звужується фільтром за протоколами Telnet та SSH, а вміст повідомлення відкривається подвійним клацанням. Місце увімкнення пристрою Sniffer студент обирає самостійно та обґрунтовує у звіті, враховуючи, що пристрій „бачить" лише той канал, у який його увімкнено. Програмний аналізатор трафіку Wireshark у складі Packet Tracer відсутній.

Середовище Packet Tracer підтримує не всі команди налагодження протоколу SSH, наведені у теоретичних відомостях. Працюють команди ip ssh version, ip ssh time-out та crypto key generate rsa; команди ip ssh authentication-retries, ip ssh rsa keypair-name, ip ssh port, ip ssh source-interface, ip ssh dh min size, ip ssh maxstartups, ip ssh dscp та ip ssh precedence відхиляються з повідомленням Invalid command. Тому ключову пару RSA слід формувати командою crypto key generate rsa із зазначенням довжини ключа, а наведений вище модельний приклад у частині іменування ключової пари (команда ip ssh rsa keypair-name) у Packet Tracer не відтворюється.

Варіанти, для яких табл. 7 передбачає протокол SSH версії 1, виконуються штатними засобами Packet Tracer (команда ip ssh version 1). Слід враховувати, що сучасні зовнішні SSH-клієнти цієї версії не підтримують – її вилучено з OpenSSH у версії 7.6 (2017 р.), тому підключення перевіряється засобами самого Packet Tracer.

1. У середовищі програмного симулятора/емулятора створити проект мережі (рис. 21). Під час побудови звернути увагу на вибір моделей комутаторів та маршрутизаторів, мережних модулів та адаптерів, а також мережних з'єднань. Для побудованої мережі заповнити описову таблицю, яка аналогічна табл. 3.

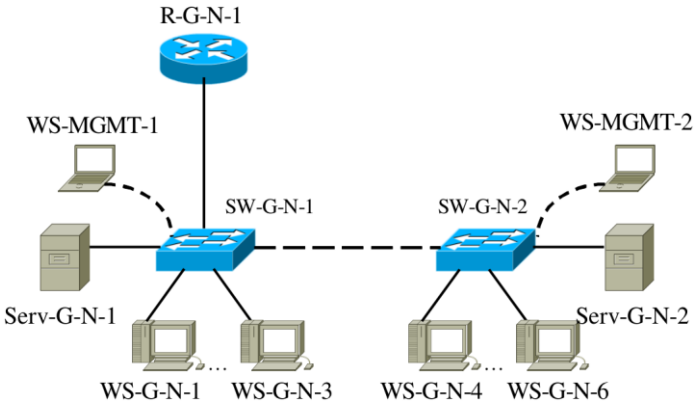


Рис. 21. Проєкт мережі

2. Розробити схему адресації пристроїв мережі. Для цього використовувати дані табл. 6. Результати навести у вигляді таблиці, яка аналогічна табл. 4.

3. Провести базове налагодження пристроїв, інтерфейсів та каналів зв'язку. Провести налагодження параметрів IP-адресації пристроїв мережі відповідно до даних, які отримані у п. 2.

4. Перевірити наявність зв'язку між всіма пристроями мережі

5. Провести налагодження віддаленого доступу до пристроїв мережі згідно з даними табл. 7 (за потреби створити користувачів на пристроях, рівень їх привілеїв встановити довільним чином)..

6. Дослідити процеси віддаленого доступу до налагоджених у п. 5 комунікаційних пристроїв. У разі відсутності доступу визначити проблеми та усунути їх.

7. Для маршрутизатора мережі, на якому налагоджено підключення з використанням засобів локальної аутентифікації на базі механізму користувачів, налагодити можливість підключення як за допомогою протоколу Telnet, так і за допомогою протоколу SSH. Дослідити можливості підключення до налагодженого пристрою за допомогою додатка PuTTY або подібного.

8. Дослідити та проаналізувати відмітності віддаленого доступу за протоколом Telnet і за протоколом SSH у розрізі передачі даних аутентифікації та передачі даних сеансу зв'язку. Для перехоплення

повідомлень використати штатні засоби програмного симулятора/емулятора.

Таблиця 6

Параметри IP-адресації мережі

№ варіанта	IP-адреса мережі A	Префікс	IP-адреса шлюзу за замовчуванням/ IP-адреса DNS-сервера
1	191.G.N.0	/24	Перша IP-адреса діапазону
2	192.G.N.0	/25	Остання IP-адреса діапазону
3	193.G.N.0	/26	Перша IP-адреса діапазону
4	194.G.N.0	/27	Остання IP-адреса діапазону
5	195.G.N.0	/28	Перша IP-адреса діапазону
6	196.G.N.0	/24	Остання IP-адреса діапазону
7	197.G.N.0	/25	Перша IP-адреса діапазону
8	198.G.N.0	/26	Остання IP-адреса діапазону
9	199.G.N.0	/27	Перша IP-адреса діапазону
10	200.G.N.0	/28	Остання IP-адреса діапазону
11	201.G.N.0	/24	Перша IP-адреса діапазону
12	202.G.N.0	/25	Остання IP-адреса діапазону
13	203.G.N.0	/26	Перша IP-адреса діапазону
14	204.G.N.0	/27	Остання IP-адреса діапазону
15	205.G.N.0	/28	Перша IP-адреса діапазону
16	206.G.N.0	/24	Остання IP-адреса діапазону
17	207.G.N.0	/25	Перша IP-адреса діапазону
18	208.G.N.0	/26	Остання IP-адреса діапазону
19	209.G.N.0	/27	Перша IP-адреса діапазону
20	210.G.N.0	/28	Остання IP-адреса діапазону
21	211.G.N.0	/24	Перша IP-адреса діапазону
22	212.G.N.0	/25	Остання IP-адреса діапазону
23	213.G.N.0	/26	Перша IP-адреса діапазону
24	214.G.N.0	/27	Остання IP-адреса діапазону
25	215.G.N.0	/28	Перша IP-адреса діапазону
26	216.G.N.0	/24	Остання IP-адреса діапазону
27	217.G.N.0	/25	Перша IP-адреса діапазону
28	218.G.N.0	/26	Остання IP-адреса діапазону
29	219.G.N.0	/27	Перша IP-адреса діапазону
30	220.G.N.0	/28	Остання IP-адреса діапазону
31	221.G.N.0	/24	Перша IP-адреса діапазону
32	222.G.N.0	/25	Остання IP-адреса діапазону
33	223.G.N.0	/26	Перша IP-адреса діапазону

Таблиця 7

Дані для вибору протоколів віддаленого доступу

№ варіанта	Протоколи віддаленого доступу		
	R-G-N-1	SW-G-N-1	SW-G-N-2
1	Telnet&Pwd	Telnet&User	SSHv1
2	Telnet&User	SSHv1	Telnet&Pwd
3	SSHv1	Telnet&Pwd	Telnet&User
4	Telnet&Pwd	Telnet&User	SSHv2
5	Telnet&User	SSHv2	Telnet&Pwd
6	SSHv2	Telnet&Pwd	Telnet&User
7	Telnet&Pwd	Telnet&User	SSHv1
8	Telnet&User	SSHv1	Telnet&Pwd
9	SSHv1	Telnet&Pwd	Telnet&User
10	Telnet&Pwd	Telnet&User	SSHv2
11	Telnet&User	SSHv2	Telnet&Pwd
12	SSHv2	Telnet&Pwd	Telnet&User
13	Telnet&Pwd	Telnet&User	SSHv1
14	Telnet&User	SSHv1	Telnet&Pwd
15	SSHv1	Telnet&Pwd	Telnet&User
16	Telnet&Pwd	Telnet&User	SSHv2
17	Telnet&User	SSHv2	Telnet&Pwd
18	SSHv2	Telnet&Pwd	Telnet&User
19	Telnet&Pwd	Telnet&User	SSHv1
20	Telnet&User	SSHv1	Telnet&Pwd
21	SSHv1	Telnet&Pwd	Telnet&User
22	Telnet&Pwd	Telnet&User	SSHv2
23	Telnet&User	SSHv2	Telnet&Pwd
24	SSHv2	Telnet&Pwd	Telnet&User
25	Telnet&Pwd	Telnet&User	SSHv1
26	Telnet&User	SSHv1	Telnet&Pwd
27	SSHv1	Telnet&Pwd	Telnet&User
28	Telnet&Pwd	Telnet&User	SSHv2
29	Telnet&User	SSHv2	Telnet&Pwd
30	SSHv2	Telnet&Pwd	Telnet&User
31	Telnet&Pwd	Telnet&User	SSHv1
32	Telnet&User	SSHv1	Telnet&Pwd
33	SSHv1	Telnet&Pwd	Telnet&User

Примітка: Telnet&Pwd – підключення за протоколом Telnet із використанням засобів локальної аутентифікації на базі механізму паролів на вхід до відповідних командних режимів; Telnet&User – підключення за протоколом Telnet із використанням засобів локальної аутентифікації на базі механізму користувачів; SSHv1, SSHv2 – підключення за протоколом SSH відповідних версій із використанням засобів локальної аутентифікації на базі механізму користувачів.

Контрольні питання

1. Поняття та призначення протоколу віддаленого доступу.
2. Основні протоколи віддаленого доступу.
3. Загальна характеристика протоколів Telnet та SSH.
4. Сфера застосування протоколів Telnet та SSH.
5. Стандартизація протоколів Telnet та SSH.
6. Характеристики протоколів Telnet та SSH стосовно моделі OSI та стеку TCP/IP.
7. Рівні протоколу SSH.
8. Характеристика рівня безпеки протоколу Telnet.
9. Характеристика рівня безпеки протоколу SSH.
10. Реалізація протоколів Telnet та SSH у сучасних ОС.
11. Реалізація протоколів Telnet та SSH провідними виробниками мережного обладнання.
12. Перелік та призначення основних команд для налагодження протоколу Telnet на пристроях Cisco.
13. Перелік та призначення основних команд моніторингу роботи протоколу Telnet на пристроях Cisco.
14. Перелік та призначення основних команд для налагодження протоколу SSH на пристроях Cisco.
15. Перелік та призначення основних команд моніторингу роботи протоколу SSH на пристроях Cisco.