Лабораторна робота № 3

Сервіс передачі файлів

1. Короткі теоретичні відомості.

File Transfer Protocol (FTP) - це стандартний мережний протокол, який використовується для обміну файлами через комп'ютерну мережу, наприклад Інтернет. FTP побудований на архітектурі клієнт-сервер і використовує окремі з'єднання для керування та передачі даних між клієнтом та сервером. Клієнтські програми спочатку були інтерактивними інструментами командного рядка зі стандартизованим синтаксисом команд, але графічні інтерфейси користувача були розроблені для всіх настільних операційних систем, що використовуються сьогодні. FTP також часто використовується як компонент програми для автоматичної передачі файлів для внутрішніх функцій програми. FTP можна використовувати з виконанням автентифікації на основі пароля користувача або анонімно.

1.1. Способи підключення.

FTP працює через протокол управління передачею TCP. Зазвичай сервери FTP прослуховують порт номер 21 (зарезервований IANA) для вхідних з'єднань від клієнтів. Підключення до цього порту з FTP-клієнта формує керуюче з'єднання, за яким передаються на передаються передаються і відповіді. FTP відкриває спеціальні з'єднання для передачі даних на інші номери портів. Параметри потоків даних залежать від конкретно режиму транспортування. Для передачі даних зазвичай використовується порт номер 20.

У активному режимі клієнт FTP відкриває порт, надсилає серверу FTP номер цього порту через з'єднання даних, і чекає з'єднання з сервера FTP. Коли FTP-сервер ініціює підключення даних до FTP-клієнта, він прив'язує вихідний порт до порту 20 на FTP-сервері.

Для того, щоб використовувати активний режим, клієнт надсилає команду PORT з IP та портом як аргументом. Формат IP та порту - "h1, h2, h3, h4, p1, p2". Кожне поле є десятковим поданням 8 бітів IP-адреси вузла, за яким слідує обраний порт даних. Наприклад, клієнт з IP-адресою 192.168.0.1, прослуховуючи порт 49154 для з'єднання даних, надішле команду "`PORT 192,168,0,1,192,2`". Поля порту слід інтерпретувати як p1 × 256 + p2 = порт (у цьому прикладі, 192 × 256 + 2 = 49154).

У пасивному режимі FTP-сервер відкриває порт, надсилає FTP-клієнту IP-адресу сервера, до якого він підключається, та порт, на якому він прослуховує (16-бітове значення, розбите на старший та молодший байти, як показано вище) і чекає з'єднання з FTP-клієнтом. У цьому випадку FTP-клієнт встановлює з'єднання даних на порт, вказаний сервером.

Для використання пасивного режиму клієнт надсилає команду PASV, на яку сервер відповість чимось подібним до "`227 Entering Passive Mode (192,168,0,1,192,52)`". Синтаксис IP-адреси та порту такий самий, як і для аргументу команди PORT.

У розширеному пасивному режимі FTP-сервер працює точно так само, як і пасивний, проте він передає лише номер порту (не розбитий на старший та молодший байти), і клієнт повинен припустити, що він підключається до тієї самої IP-адреси сервера, до якої встановлено керуюче з'єднання. Розширений пасивний режим був доданий RFC 2428 у вересні 1998 року.

Поки дані передаються через потік даних, потік керування не працює. Це може спричинити проблеми з великою передачею даних через брандмауери, які очікують сеанси після тривалих періодів простою. Хоча файл цілком може бути успішно переданий, брандмауер може відключити сеанс керування, що спричинить генерування помилки.

Протокол FTP підтримує відновлення перерваних завантажень за допомогою команди REST. Клієнт передає кількість байтів, яку він вже отримав як аргумент, команді REST і перезапускає передачу. Наприклад, у деяких клієнтах командного рядка існує часто ігнорувана, але цінна команда "reget" (що означає "отримати знову"), яка призведе до продовження перерваної команди "get".

Відновити завантаження не так просто. Хоча протокол FTP підтримує команду APPE для додавання даних до файлу на сервері, клієнт не знає точної позиції, в якій перервана передача. Він повинен отримати розмір файлу іншим способом, наприклад, за допомогою списку каталогів або за допомогою команди SIZE.

У режимі ASCII (див. Нижче) відновлення передачі може бути проблематичним, якщо клієнт і сервер використовують різні символи кінця рядка.

1.2. Проблеми з безпекою.

Оригінальна специфікація FTP є за своєю суттю незахищеним способом передачі файлів, оскільки не існує жодного способу передачі даних і команд в зашифрованому вигляді. Це означає, що в більшості мережних конфігурацій імена користувачів, паролі, команди FTP та передані файли можуть бути захоплені будь-якою людиною за допомогою сніфера пакетів. Це проблема, загальна для багатьох специфікацій протоколів Інтернету, написаних до створення рівня захищених сокетів (SSL), таких як HTTP, SMTP та Telnet. Загальним рішенням цієї проблеми є використання SFTP (протокол передачі файлів SSH) або FTPS (FTP поверх SSL), який додає до FTP шифрування SSL або TLS, як зазначено в RFC 4217.

1.3. Коди повернення FTP.

Коди повернення сервера FTP вказують їх статус цифрами, що знаходяться в них. Коротке пояснення значень різних кодів наведено нижче:

- 1xx: Позитивна попередня відповідь. Запрошена дія ініціюється, але перед її початком буде надіслана інша відповідь.
- 2xx: Позитивна відповідь на завершення. Запитану дію завершено. Клієнт тепер може виконати нову команду.
- 3xx: Позитивна проміжна відповідь. Команда була успішною, але необхідна подальша команда, перш ніж сервер зможе діяти за запитом.
- 4xx: Перехідна відповідь негативного завершення. Команда не була успішною, але клієнт може вільно спробувати команду ще раз, оскільки помилка є лише тимчасовою.
- 5xx: Постійна негативна відповідь. Команда не була успішною, і клієнт не повинен намагатися повторити її ще раз.
- x0x: Помилка сталася через синтаксичну помилку.
- x1x: Ця відповідь є відповіддю на запит на інформацію.
- x2x: Ця відповідь є відповіддю, що стосується інформації про з'єднання.
- x3x: Ця відповідь є відповіддю, що стосується обліку та авторизації.
- x4x: поки не вказано

- x5x: ці відповіді вказують на стан файлової системи сервера щодо запитуваної передачі чи іншої дії файлової системи.

1.4. Анонімний FTP.

Вузол, який надає послугу FTP, може додатково забезпечити анонімний доступ до FTP. Зазвичай користувачі входять до служби за допомогою анонімного облікового запису, коли з'являється запит на ім'я користувача. Хоча користувачів зазвичай просять надіслати свою електронну адресу замість пароля, фактично перевірка наданих даних не виконується.

Оскільки сучасні FTP-клієнти зазвичай приховують анонімний процес входу від користувача, ftp-клієнт надасть фіктивні дані на запит пароля (оскільки адреса електронної пошти користувача може бути невідома застосунку). Наприклад, такі агенти користувача ftp вказують перелічені паролі для анонімних входів:

- Mozilla Firefox (3.5.2) - `mozilla@example.com`
- KDE Konqueror (3.5) - `anonymous@`
- wget (1.10.2) - `wget@`
- lftp (3.4.4) - `lftp@`
- Opera (9.6.4) - `opera@`

Протокол Gopher пропонується як альтернатива анонімному FTP, а також протокол службових файлів.

1.5. Параметри передачі.

Відповідно до стандарту FTP RFC959 передача даних визначається чотирма основними параметрами:

- структура даних: орієнтована на потік, на запис або на сторінку
- тип даних: текстові типи ASCII, EBCDIC, з підтипами для різних варіантів керування кареткою; бінарні типи орієнтовані на байт або довільну довжину слів
- контроль вертикального формату: для текстових типів ASCII та EBCDIC, якщо вказано вертикальне управління форматом
- режим передачі: передача, орієнтована на потік, нестиснута передача, орієнтована на блок, або передача, орієнтована на стиснений блок

До 1990-х років використання FTP зосереджувалося на структурі файлів, орієнтованих на потік, та режимі передачі, орієнтованих на потік; більшість FTP-серверів та клієнтів, починаючи з 1990-х років, не підтримують інші файлові структури або режими передачі.

1.6. Структура даних.

Структура даних визначається за допомогою команди STRU. Наступні файлові структури визначені в розділі 3.1.1 RFC959:

- Структура F або FILE (орієнтована на потік). Файли розглядаються як довільна послідовність байтів, символів або слів. Це звичайна структура файлів у системах Unix та інших системах, таких як CP/M, MSDOS та Microsoft Windows. [Розділ 3.1.1.1]
- R або RECORD структура (орієнтована на запис). Файли розглядаються як розділені на записи, які можуть мати фіксовану або змінну довжину. Ця організація файлів є загальною для систем мейнфреймів та середнього класу, таких як MVS, VM/CMS, OS/400 та VMS.

- P або PAGE структура (орієнтована на сторінку). Файли розділені на сторінки, які можуть містити дані або метадані; кожна сторінка може також мати заголовок із різними атрибутами. Ця файлова структура була спеціально розроблена для систем TENEX і, як правило, не підтримується на інших платформах. RFC1123, розділ 4.1.2.3, рекомендує не застосовувати цю структуру.

1.7. Типи даних.

Тип даних визначається за допомогою команди TYPE. Визначено такі типи даних:

- A (ASCII). Текстові дані, передані через мережу в наборі символів NVT ASCII.
- E (EBCDIC). Текстові дані, передані через мережу в наборі символів EBCDIC.
- I або IMAGE (орієнтований на байти). Двійкові дані передаються у вигляді потоку 8-бітових байтів.
- L або LOCAL (орієнтований на слова). Двійкові дані, передані як потік слів. Кількість бітів у слові вказується як аргумент, наприклад: L32 для 32-розрядних слів, L36 для 36-розрядних слів.

Історично поширеною проблемою були клієнти та сервери FTP, які за замовчуванням мають тип ASCII, але не забезпечують жодного захисту від передачі двійкових файлів. В результаті двійкові файли пошкоджені, наприклад, заміна символів нового рядка. У більшості сучасних клієнтів цього можна уникнути, якщо автоматично встановити тип зображення. Іншим підходом буде вибір FTP TYPE на основі типу файлу, записаного у файловій системі (для тих файлових систем, які роблять це), або евристично.

L8 фактично еквівалентний I, і більшість FTP-серверів або клієнтів не приймають інші розміри слів, крім 36-розрядних платформ. Дані повинні бути передані в упакованому двійковому форматі.

Зверніть увагу, тип даних вказує тип для передачі, а не тип, в якому дані зберігаються в системах відправника або отрмувача. Клієнт та сервер можуть вільно перетворювати дані у форму, яка є найбільш зручною на їх платформі. Наприклад, текстові типи даних A та E можуть піддаватися трансляції набору символів (наприклад, ASCII проти EBCDIC), замін символів нового рядка (наприклад, CRLF проти LF) або трансляція текстових даних між орієнтованими на потік та орієнтованими на запис форматами (тобто один запис на рядок, можливо, заповнений пробілами до максимальної довжини рядка проти орієнтованого на потік символами нового рядка для розділення рядків).

Часто FTP-клієнти використовують слово "MODE" для позначення типу даних, хоча це помилково, оскільки слово "MODE" вже прийнято для позначення режиму передачі.

1.8. Режим передачі.

Режим передачі визначається командою MODE. Визначаються наступні режими:

- S або STREAM MODE: дані представлені у вигляді потоку 8-бітових байтів. Для файлів, орієнтованих на запис, визначено механізм екранування, який чітко вказує межі записів та явний кінець файлу. Для потоково-орієнтованих файлів не визначено жодного механізму екранування, а кінець файлу представлений закриттям з'єднання.
- B або BLOCK MODE: дані представлені у вигляді потоку блоків. Кожен блок має заголовок, який вказує його довжину, а також прапори для позначення кінця запису та кінця файлу. Прапори також можуть використовуватися для позначення підозрілого блоку даних, наприклад блок даних, зчитуваний з магнітної стрічки, який мав помилку

контрольної суми, але все одно передається, хоча він може містити спотворені дані. Також підтримує маркери перезапуску, які дозволяють перезапустити передачу даних з цієї точки.

- C або COMPRESSED MODE: подібний до потокового режиму, але додає підтримку кодування довжини циклу, а також прапори, визначені в блочному режимі.

Зараз більшість FTP-клієнтів та серверів підтримують лише режим STREAM.

1.9. FTP команди.

Команди, які починаються з літери X, зазвичай зарезервовані для експериментальних розширень, хоча замість цього слід використовувати підкоманди SITE.

RFC959 визначає такі команди FTP, які також були присутні в RFC765:

- USER: надає ім’я користувача для входу.
- PASS: надає пароль для входу.
- ACCT: надає облікову інформацію. Наприклад, користувач може працювати над кількома проектами; обліковий запис може бути використаний для того, щоб стягнути плату за зберігання даних за правильним проектом. (Зазвичай не застосовується).
- CWD: змінює робочий каталог на вказаний.
- REIN: видаляє всю інформацію про автентифікацію та параметри; має відбуватися повторна реєстрація через USER.
- QUIT: розриває з'єднання.
- PORT: вказує вузол/порту для передачі даних.
- PASV: увійти в пасивний режим.
- TYPE: вказує тип даних та вертикальний контроль формату.
- STRU: вказує структуру даних.
- MODE: вказує режим передачі.
- RETR: ініціює передачу даних із сервера на клієнт із зазначенням імені файлу для отримання.
- STOR: ініціює передачу даних з клієнта на сервер, вказавши ім'я файлу, який слід зберігати на сервері.
- APPE: подібно до STOR, за винятком того, що файл вже існує, додає отримані дані до кінця, а не створює новий.
- ALLO: виділяє місце для файлу. Необов’язково вказує максимальний розмір кожного запису.
- REST: визначає маркер перезапуску, з якого слід відновити передачу. Спочатку призначений для використання з маркерами перезапуску, надісланими сервером у режимі B або C, але згодом розширений у RFC3659 до зміщення байтів, зазначених у режимі S.
- RNFR: щоб перейменувати файл, вкажіть файл, який буде перейменовано.
- RNTO: щоб перейменувати файл, вказує нову назву файлу та виконує перейменування. Часто також використовується для переміщення файлів.
- DELE: видаляє файл.
- PWD: друкує поточний робочий каталог.
- LIST: відкриває з'єднання даних із типом даних A або E для передачі списку файлів у поточному каталозі. Формат даних специфічний для системи, але призначений для читання людиною.

- NLST: подібний до LIST, але передає без розмітки імена файлів за допомогою CRLF або NL.
- SITE: надає підкоманди для виконання певних системних послуг. Характер цих послуг не визначений.
- STAT: без аргументів, поточний стан з'єднання. З аргументом, еквівалентним LIST, але список передається через контрольне з'єднання, інкапсульоване в повідомленнях.
- HELP: надає довідку, необов’язково аргумент для вказівки конкретної команди, за якою запитується допомога.
- NOOP: нічого не робить.

RFC959 додає такі нові команди, яких не було в RFC765:

- CDUP: змінює робочий каталог на батьківський. В даний час позначення батьківського каталогу різняться залежно від платформи (хоча найчастіше .. в системах, що походять з Unix або MS DOS).
- SMNT: підключіть іншу файлову систему або том. Призначений для систем, таких як DOS або VMS, де існує різниця між томом та каталогом у іменах шляхів; але зазвичай не виконуються навіть на таких системах.
- STOU: унікальний завантаження на сервер - ініціює передачу даних з клієнта на сервер; сервер повинен вибрати унікальне ім'я для файлу, який потрібно отримати.
- RMD: видаляє каталог.
- MKD: створює каталог.
- SYST: визначає операційну систему сервера.

RFC765 описав ряд команд, які були видалені в RFC959. Вони не були частиною реалізацій FTP з початку 1980-х, оскільки їх функціональність пізніше (частково) була замінена SMTP:

- MLFL: використовується для надсилання електронної пошти через з'єднання для передачі даних.
- MAIL: використовується для надсилання електронної пошти через контрольне з'єднання.
- MSND: як MAIL, але надсилає дані безпосередньо на термінал користувача, а не на їх поштову скриньку.
- MSOM: поводиться як MAIL або MSND - надсилати на термінал, якщо це дозволено, інакше на поштову скриньку.
- MSAM: подібний до MSOM - за винятком того, що MSOM надсилає на поштову скриньку лише в тому випадку, якщо доставка на термінал неможлива; але MSAM надсилає на поштову скриньку незалежно від того, чи успішно здійснена спроба доставки на термінал.
- MRSQ: дозволяє передавати одне електронне повідомлення декільком користувачам на одному вузлі.
- MRCP: після MRSQ ідентифікує одного з таких одержувачів; повторюється для кожного одержувача.

RFC2228 додає ряд команд, пов’язаних із шифруванням та автентифікацією повідомлень:

- AUTH: визначає механізм автентифікації/захисту, який буде використовуватися.
- ADAT: визначає дані безпеки, характерні для обраного механізму AUTH.
- PBSZ: використовується для узгодження максимального розміру буфера для зашифрованих даних.
- PROT: визначає рівень захисту каналу даних. Визначені такі рівні:
  - C (Clear) - канал передачі даних не підлягає ні шифруванню, ні захисту цілісності.

    - S (Safe) - захист цілісності, що застосовується до каналу даних.
    - E (конфіденційно) - шифрування, що застосовується до каналу даних.
    - P (Private) - як шифрування, так і захист цілісності, що застосовуються до каналу даних.
- CCC: вимикає захист цілісності для наступних команд в каналі управління.
- MIC: надсилає команду із захистом цілісності.
- CONF: надсилає команду із захистом конфіденційності.
- ENC: надсилає команду із захистом цілісності та конфіденційності.

RFC1639 додає підтримку FTP через довільні транспортні протоколи, такі як IPX/SPX. Для цього він визначає дві нові команди:

- LPRT: схожий на PORT, але підтримує довільні формати адрес та портів.
- LPSV: подібне розширення до PASV.

RFC2389 визначає дві нові команди, що використовуються як загальний механізм розширення для FTP:

- FEAT: отримує список додаткових функцій, що підтримуються FTP-сервером.
- OPTS: загальний механізм для клієнта для вказівки параметрів довільних команд FTP.

RFC2428 додає дві нові команди, подібні в принципі до RFC1639, але різні в деталях:

- EPRT: подібний до PORT, але підтримує довільні сімейства адрес, а не лише IPv4; спеціально призначений для IPv6.
- EPSV: подібне розширення до PASV

LPRT надсилає адреси у вигляді довільного рядка-октету (хоча і в десятковому кодуванні), EPRT - у форматованому рядку, формат рядка залежить від формату адреси. EPRT передбачає використання 16-розрядних номерів портів у стилі TCP, тоді як LPRT є більш гнучким і підтримує транспортні протоколи з номерами портів, що перевищують 16-бітові.

RFC2640 додає одну нову команду:

- LANG: використовується для вибору мови для FTP-повідомлень

RFC3659 визначає кілька нових команд:

- MDTM: отримує час модифікації файлу
- SIZE: отримати розмір файлу
- MLSD: отримання списку файлів у каталозі. На відміну від NLST, повертає не тільки імена файлів, а й атрибути; але на відміну від LIST, він повертає атрибути у розширюваному стандартизованому форматі, а не в довільному, специфічному для платформи.
- MLST: те саме, що MLSD, але отримує список для окремого файлу, а не каталогу. Для каталогів отримує власні атрибути, а не їх перелік. MLST не вимагає з'єднання даних, але повертає один рядок, що містить список для запитуваного шляху.

1.10. FTP і NAT.

Представлення IP-адрес та номерів портів у команді PORT та відповіді PASV створює ще одну проблему для пристроїв трансляції мережних адрес (NAT) при обробці FTP. Пристрій NAT має змінити ці значення, щоб вони містили IP-адресу NAT-клієнта та порт, вибраний NAT-пристроєм для з'єднання даних. Нова адреса та порт, можливо, за своєю десятковою подачею відрізнятимуться від початкової адреси та порту. Це означає, що зміна значень на контрольному підключенні пристроєм NAT повинна виконуватися обережно, змінюючи поля TCP Sequence і

Acknowledgment для всіх наступних пакетів. Така трансляція зазвичай не виконується в більшості пристроїв NAT, але для цього існують спеціальні шлюзи прикладного рівня.

1.11. FTP через SSH.

FTP через SSH іноді називають безпечним FTP; його не слід плутати з іншими методами захисту FTP, такими як SSL/TLS (FTPS). Інші методи передачі файлів за допомогою SSH, які не пов'язані з FTP, включають SFTP та SCP; у кожному з них весь обмін (облікові дані та дані файлів) завжди захищена протоколом SSH.

2. Завдання на роботу.

2.1. Створити закритий ключ та запит на сертифікат для доменного імені, яке відповідає варіанту завдання. Підписати запит на сертифікат, використовуючи сертифікат і закритий ключ центру сертифікації (CA), створеного в лабораторній роботі № 2.

2.2. Встановити та налаштувати файловий сервер, який реалізує протоколи FTP, FTPS та SFTP та відповідає наступним вимогам:

- анонімний доступ (користувач ftp або anonymous);
- в домашньому каталозі анонімного користувача створені 2 каталоги: pub та incoming, каталог pub має права тільки читання, каталог incoming має права на читання та запис;
- створені користувачі з іменами, доступом та правами, які відповідають варіанту завдання;
- для створених користувачів забезпечується доступ по протоколам, які відповідають варіанту завдання.

2.3. Виконати аналіз протокольного обміну між клієнтом та сервером під час автентифікації та пересилки файлів.

2.4. Рекомендується використовувати наступне програмне забезпечення:

- FTP-сервер: vsftpd + openssh або proftpd;
- FTP-клієнт: lftp.

2.5. Для перевірки роботи файлового серверу та аналізу протокольного обміну рекомендується використовувати утиліти: ftp, telnet, netcat, openssl, lftp.

2.6. Додати запис типу A для доменного імені файлового серверу в DNS. Додати сертифікат власного CA у список довірених на вузлі, де запускається FTP-клієнт.

Варіант	**Ім'я FTP-серверу**	**Користувачі**			
		Ім'я	**Протокол**	**Доступ**	**Права**
1	ftp.letter.net	alpha	FTP,FTPS	до всього дерева каталогів	RW
		beta	SFTP	тільки домашній каталог	R
		gamma	FTP,FTPS,SFTP	тільки домашній каталог	RW
		delta	SFTP	до всього дерева каталогів	RW
		omega	FTP,FTPS	тільки домашній каталог	RW

Варіант	Ім'я FTP-серверу	Користувачі			
		Ім'я	Протокол	Доступ	Права
2	ftp.planet.edu	mercury	FTP,FTPS	тільки домашній каталог	R
		venus	SFTP	до всього дерева каталогів	R
		earth	FTP,FTPS,SFTP	тільки домашній каталог	RW
		saturn	SFTP	тільки домашній каталог	R
		jupiter	FTP,FTPS	до всього дерева каталогів	RW
3	ftp.cat.com	tiger	FTP,FTPS	до всього дерева каталогів	R
		lion	SFTP	тільки домашній каталог	RW
		lynx	FTP,FTPS,SFTP	тільки домашній каталог	RW
		leopard	SFTP	до всього дерева каталогів	RW
		jaguar	FTP,FTPS	тільки домашній каталог	R
4	ftp.flower.org	rose	FTP,FTPS	тільки домашній каталог	RW
		gerbera	SFTP	до всього дерева каталогів	RW
		tulip	FTP,FTPS,SFTP	тільки домашній каталог	R
		aster	SFTP	тільки домашній каталог	RW
		peony	FTP,FTPS	до всього дерева каталогів	R
5	ftp.linux.net	ubuntu	FTP,FTPS	до всього дерева каталогів	RW
		debian	SFTP	тільки домашній каталог	R
		centos	FTP,FTPS,SFTP	тільки домашній каталог	R
		gentoo	SFTP	до всього дерева каталогів	R
		fedora	FTP,FTPS	тільки домашній каталог	RW
6	ftp.color.edu	red	FTP,FTPS	тільки домашній каталог	R
		green	SFTP	до всього дерева каталогів	RW
		blue	FTP,FTPS,SFTP	тільки домашній каталог	RW
		black	SFTP	тільки домашній каталог	R
		white	FTP,FTPS	до всього дерева каталогів	RW
7	ftp.metal.com	gold	FTP,FTPS	до всього дерева каталогів	RW
		silver	SFTP	тільки домашній каталог	RW
		iron	FTP,FTPS,SFTP	тільки домашній каталог	R
		copper	SFTP	до всього дерева каталогів	R
		zinc	FTP,FTPS	тільки домашній каталог	RW

Варіант	Ім'я FTP-серверу	Користувачі			
		Ім'я	Протокол	Доступ	Права
8	ftp.capital.org	london	FTP,FTPS	тільки домашній каталог	RW
		tokyo	SFTP	до всього дерева каталогів	R
		paris	FTP,FTPS,SFTP	тільки домашній каталог	RW
		rome	SFTP	тільки домашній каталог	R
		berlin	FTP,FTPS	до всього дерева каталогів	RW
9	ftp.currency.net	dollar	FTP,FTPS	до всього дерева каталогів	R
		dinar	SFTP	тільки домашній каталог	R
		lira	FTP,FTPS,SFTP	тільки домашній каталог	RW
		peso	SFTP	до всього дерева каталогів	RW
		real	FTP,FTPS	тільки домашній каталог	R
10	ftp.river.edu	nile	FTP,FTPS	тільки домашній каталог	RW
		amazon	SFTP	до всього дерева каталогів	RW
		congo	FTP,FTPS,SFTP	тільки домашній каталог	RW
		amur	SFTP	тільки домашній каталог	R
		mekong	FTP,FTPS	до всього дерева каталогів	R
11	ftp.fruit.com	apple	FTP,FTPS	до всього дерева каталогів	RW
		orange	SFTP	тільки домашній каталог	R
		grape	FTP,FTPS,SFTP	тільки домашній каталог	R
		banana	SFTP	до всього дерева каталогів	R
		lemon	FTP,FTPS	тільки домашній каталог	RW
12	ftp.digit.org	one	FTP,FTPS	тільки домашній каталог	RW
		two	SFTP	до всього дерева каталогів	R
		three	FTP,FTPS,SFTP	тільки домашній каталог	R
		four	SFTP	тільки домашній каталог	RW
		five	FTP,FTPS	до всього дерева каталогів	RW
13	ftp.month.net	march	FTP,FTPS	до всього дерева каталогів	R
		april	SFTP	тільки домашній каталог	RW
		may	FTP,FTPS,SFTP	тільки домашній каталог	RW
		june	SFTP	до всього дерева каталогів	RW
		july	FTP,FTPS	тільки домашній каталог	R

Варіант	Ім'я FTP-серверу	Користувачі			
		Ім'я	Протокол	Доступ	Права
14	ftp.name.edu	maria	FTP,FTPS	тільки домашній каталог	RW
		tomas	SFTP	до всього дерева каталогів	R
		tereza	FTP,FTPS,SFTP	тільки домашній каталог	RW
		stefan	SFTP	тільки домашній каталог	R
		sara	FTP,FTPS	до всього дерева каталогів	RW
15	ftp.country.com	france	FTP,FTPS	до всього дерева каталогів	RW
		china	SFTP	тільки домашній каталог	RW
		spain	FTP,FTPS,SFTP	тільки домашній каталог	R
		italy	SFTP	до всього дерева каталогів	RW
		germany	FTP,FTPS	тільки домашній каталог	R

3. Контрольні питання.

3.1. Протокол FTP призначення та особливості роботи.

3.2. Пасивний та активний режими роботи FTP-сервера.

3.3. Команди протоколу FTP.

3.4. Протокол FTPS (FTP зверху SSL/TLS).

3.5. Протокол SFTP.

4. Література.

https://web.archive.org/web/20210410193501/http://wiki.gis.com/wiki/index.php/File_Transfer_Protocol

RFC959 https://tools.ietf.org/html/rfc959

SSH File Transfer Protocol https://tools.ietf.org/html/draft-ietf-secsh-filexfer-13