SSH и FTP

**Предмет: Операционни системи**

Симеон Марков, 11А

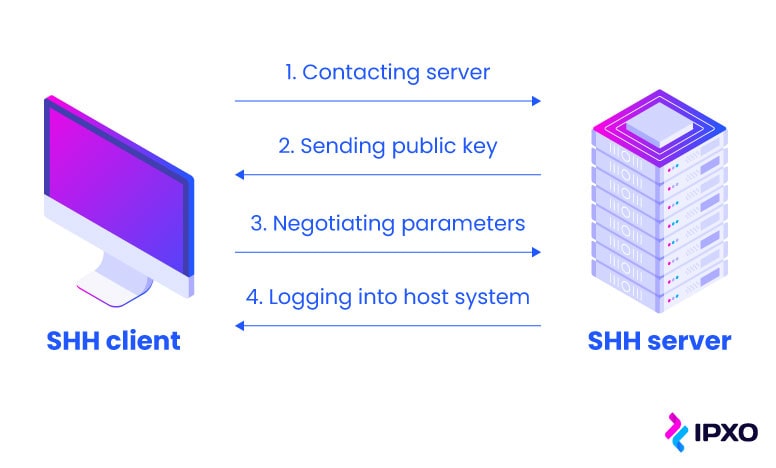
2024

1. SSH (Secure Shell)

* SSH, или Secure Shell, е протокол, който осигурява сигурна, криптирана комуникация между два компютъра.
* Обикновено се използва за отдалечено влизане, дистанционно изпълнение на команди и прехвърляне на файлове.
* SSH е проектиран като заместител на по-малко сигурния протокол Telnet, който предава данни в обикновен текст и е податлив на подслушване.
* SSH използва криптиране, за да защити данните, които се предават по мрежата.
* Когато потребител влезе в отдалечена система с помощта на SSH, потребителското име и паролата на потребителя се криптират и изпращат до отдалечената система.
* След това отдалечената система проверява идентификационните данни на потребителя и установява сигурна връзка между двете системи.
* След като се установи сигурна връзка, потребителят може да взаимодейства с отдалечената система.

1. Структура на SSH
   * + **Transport layer protocol (TLP)**: служи за удостоверяване на сървъра и установяване на поверителност и цялост.
     + **User authentication protocol:** удостоверява потребителя на сървъра, потвърждавайки самоличността на агента, действащ като клиент.
     + **Connection protocol:** мултиплексира SSH тунела(създава отделни потоци от данни или логически канали от една връзка клиент/сървър.)
2. Как работи сървърът SSH ?

* SSH протоколът работи на модела **клиент/сървър**. SSH клиентът винаги инициира настройката на защитената връзка, а SSH сървърът слуша за входящи заявки за връзка (обикновено на TCP порт 22 на хост системата) и отговаря на тях.
* Във фазата на настройка на връзката SSH сървърът се удостоверява пред клиента, като предоставя своя публичен ключ. Това позволява на SSH клиента да провери дали всъщност комуникира с правилния SSH сървър (вместо с нападател, който може да се представи за сървър).
* След успешно удостоверяване, сървърът предоставя на клиента достъп до хост системата. Този достъп се управлява от разрешенията за потребителски акаунт в целевата хост система.
* Автоматизираните SSH сесии много често се използват като част от много автоматизирани процеси, които изпълняват задачи като събиране на регистрационни файлове, архивиране, мрежови архиви и други критични задачи на системно ниво.



1. Силно удостоверяване със SSH ключове
   * + - Няколко опции могат да се използват за удостоверяване на потребителя. Най-често срещаните са **пароли** и удостоверяване с **публичен ключ.**
       - Методът за удостоверяване с публичен ключ се използва предимно за автоматизация.
       - Идеята е да има двойка криптографски ключове - публичен ключ и частен ключ - и да конфигурира публичния ключ на сървър, за да оторизира достъпа и да предостави на всеки, който има копие на частния ключ, достъп до сървъра.
2. Използване на SSH
3. Осигурява сигурен достъп на потребителите и автоматизирани процеси.
4. Лесен и сигурен начин за прехвърляне на файлове от една система в друга през несигурна мрежа.
5. Също така издава отдалечени команди на потребителите.
6. помага на потребителите да управляват мрежовата инфраструктура и други критични компоненти на системата.
7. Използва се за влизане в shell на отдалечена система (Host), която замества Telnet и rlogin и се използва за изпълнение на една команда на хоста, която замества rsh.
8. Комбинира се с помощната програма rsync за архивиране, копиране и огледално отразяване на файлове с пълна сигурност и ефективност.
9. Може да се използва за препращане на порт.
10. С помощта на SSH можем да настроим автоматичното влизане в отдалечен сървър като **OpenSSH**.
11. Можем сигурно да сърфираме в мрежата чрез криптираната прокси връзка със SSH клиента, поддържайки протокола SOCKS.
12. Защо SSH ?
    * + Традиционните механизми за удостоверяване, базирани на пароли, вече не работят и където и да присъстват, те са източниците на мъгла и триене, което води до многобройни нарушения на данните или инциденти поради злоупотреба с идентификационни данни.
13. Как се активира SSH на Windows ?
14. Отворете "Настройки" > "System" > "Допълнителни функции" и проверете дали клиентът на **OpenSSH** е инсталиран – ако не е, инсталирайте го. За повече информация прочетете [[[**тук**](https://learn.microsoft.com/en-us/windows-server/administration/openssh/openssh_install_firstuse?tabs=gui#install-openssh-for-windows)](https://learn.microsoft.com/en-us/windows-server/administration/openssh/openssh_install_firstuse?tabs=gui#install-openssh-for-windows)**.**](https://learn.microsoft.com/en-us/windows-server/administration/openssh/openssh_install_firstuse?tabs=gui#install-openssh-for-windows)

A screenshot of a computer

Description automatically generated

1. Отворете командния ред като администратор.
2. Въведете командата ssh domain\username@servername.
3. Ако е необходимо, помислете за порта: по подразбиране е порт 22, променете порта с -pPortnumber зад името на сървъра.
4. Ако е необходимо, потвърдете пръстовия отпечатък на хост ключа с Да и въведете парола за първата връзка.
5. FTP (File Transfer Protocol)

* мрежов протокол за предаване на файлове между компютри чрез Transmission Control Protocol/Internet Protocol **(TCP/IP)** връзки. В рамките на TCP/IP пакета, FTP се счита за протокол на приложния слой.
* Потребителите се нуждаят от интернет връзка, за да изпълняват FTP трансфери.

1. Архитектура клиент-сървър

* **Клиент:** Клиентът е устройството или софтуерът на потребителя, който инициира FTP връзката със сървъра.
* **Сървър:** Сървърът е отдалечената система, която хоства FTP услугата и управлява прехвърлянето на файлове.

1. Как работи FTP ?

* Работи по модел **клиент-сървър.**
* Връзките за прехвърляне на файлове (обикновено) се инициират от **FTP клиент** и се отговаря от **FTP сървър**.
* След като се установи удостоверена връзка между клиента и сървъра, файловете и папките могат да бъдат прехвърлени в двете посоки между двата свързани компютъра.

A diagram of a diagram

Description automatically generated

1. За какво се изпозва FTP сървър

* Обмен на файлове с голям размер
* Оптимизиране на работни потоци
* Подобряване на контрола
* Надеждно възстановяване след злополука
* Репликация
* Достъп и зареждане на данни

1. Видове FTP
2. **Anonymous FTP**: е активиран на някои сайтове, чиито файлове са достъпни за публичен достъп.
3. **Password Protected FTP**: подобен на предишния, но промяната в него е използването на потребителско име и парола. В този случай данните също не са криптирани.
4. **FTP Secure (FTPS):** Всеки път, когато е установена FTP връзка, защитата на транспортния слой (TLS) е включена.
5. **FTP over Explicit SSL/TLS (FTPES**): FTPES помага чрез надстройване на FTP връзка от порт 21 до криптирана връзка.
6. **Secure FTP (SFTP):** SFTP не е FTP протокол, но е подмножество на Secure Shell Protocol, тъй като работи на порт 22.

`

1. Типове FTP сървъри

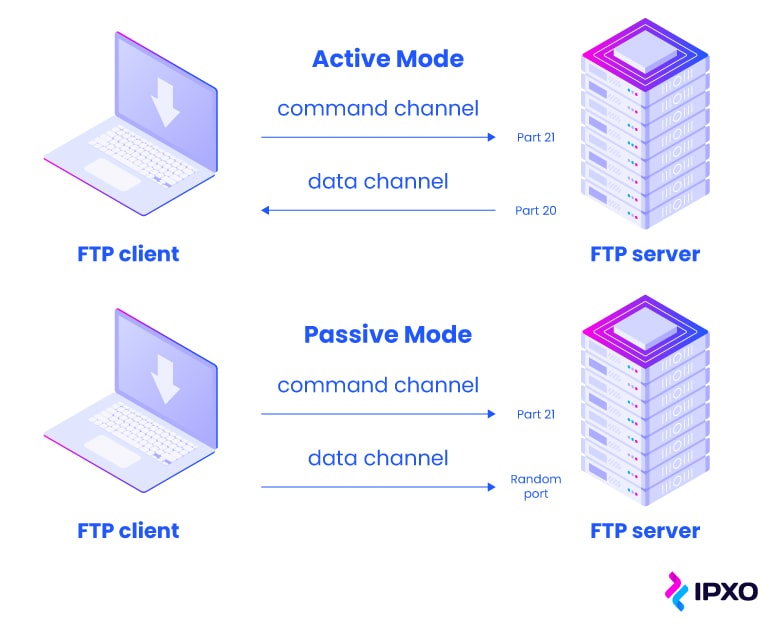
* **Anonymous Server:** Анонимният сървър е общ FTP сървър, който е за всички FTP клиенти. За достъп до този сървър не е необходима парола. Повечето FTP клиенти го използват.
* **Non-Anonymous Server:** Сървърът, който не е анонимен, е платен сървър. Ако потребителят използва неанонимен сървър, потребителят се нуждае от паролата за достъп до файла.

1. Типове връзки в FTP

* **Control Connection:** За изпращане на контролна информация като идентификация на потребителя, парола, команди за промяна на отдалечената директория, команди за извличане и съхраняване на файлове и т.н..
* **Data connection:** За изпращане на действителния файл.

1. FTP сесии

* **Активен режим:** След като клиентът инициира сесия чрез заявка за команден канал, сървърът създава връзка за данни обратно към клиента и започва да прехвърля данни.
* **Пасивен режим:** Сървърът използва командния канал, за да изпрати на клиента информацията, от която се нуждае, за да отвори канал за данни.



1. Защо да използваме FTP сървър

* Website Hosting
* Automation and Scripting
* Collaboration
* Automated Data Distribution
* Cross-Platform Compatibility

1. Роля на FTP сървъра

* FTP сървърът позволява изтеглянето и качването на файлове.
* Администраторът на FTP сървъра може да ограничи достъпа за изтегляне на различни файлове и от различни папки, пребиваващи в FTP сървъра.

1. Ниво на сигурност при FTP
2. **Данни, изпратени в чист текст:** Една от най-съществените слабости в сигурността на FTP е, че както данните, които се прехвърлят, така и потребителските идентификационни данни (потребителско име и парола) се изпращат в ясен текст.
3. **Няма шифроване:** FTP клиентите по своята същност не поддържат криптиране. Това означава, че данните не се бъркат по време на предаването, което ги прави уязвими за прихващане и излагане.
4. **Анонимен достъп:** FTP сървърите често позволяват анонимен достъп, което означава, че всеки може да се свърже с тях.
5. **Уязвимост на атаките с груба сила:** Тъй като FTP изисква потребителско име и парола, известни също като FTP клиент, за достъп до сървъра, той е податлив на атаки с груба сила.
6. **Подправяне и отвличане на сесия:** FTP също е уязвим за подправяне на атаки, при които злонамерена страна се представя за сървър или клиент, за да получи неоторизиран достъп.
7. Алтернативи на FTP
   * + SFTP (SSH File Transfer Protocol)
     + FTPS (File Transfer Protocol over SSL/TLS)
     + AS2 (Applicability Statement 2)
     + HTTPS (Hypertext Transfer Protocol Secure)
     + MFT (Managed File Transfer)
8. Клиенти на FTP

* FTP клиентите се използват за качване, изтегляне и управление на файлове на сървър. FTP клиентите включват:

1. [FileZilla](https://www.techtarget.com/whatis/definition/FileZilla)
2. [Transmit](https://panic.com/transmit/)
3. [WinSCP](https://winscp.net/eng/download.php)
4. [WS\_FTP](https://www.ipswitch.com/ftp-client)
5. FTP команди

* **USER** - изпраща идентификацията на потребителя на сървъра.
* **PASS** - изпраща потребителската парола на сървъра.
* **CWD** - позволява на потребителя да работи с различна директория или набор от данни
* **RMD** - Става причина директорията, зададена в името на пътя, да бъде премахната като директория.
* **MKD** - Става причина директорията, зададена в името на пътя, да бъде създадена като директория.
* **PWD** - Става причина името на текущата работна директория да бъде върнато в отговора.
* **RETR** - кара отдалечения хост да започне връзка с данни и да изпрати искания файл по връзката с данни.
* **STOR** - съхранява файл в текущата директория на отдалечения хост.
* **LIST** - Изпраща искане за показване на списъка с всички файлове, присъстващи в директорията.
* **ABOR** - казва на сървъра да прекрати предишната команда FTP услуга и всяко свързано прехвърляне на данни.
* **QUIT** - прекратява ПОТРЕБИТЕЛ и ако прехвърлянето на файлове не е в ход, сървърът затваря контролната връзка.

1. FTP отговори

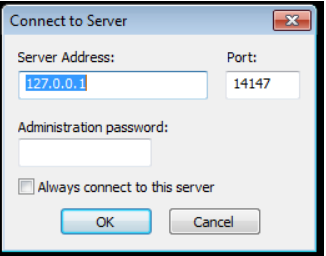
* **200** – команда е приета.
* **530** – не е влязъл.
* **331** - Потребителско име добре, трябва парола.
* **225** - Отворена връзка за данни; Не се извършва прехвърляне.
* **221** - Услугата затваря контролната връзка.
* **551** - Исканото действие е прекратено: неизвестен тип страница.
* **502** - Командата не е реализирана.
* **503**  - Лоша последователност от команди.
* **504** - Командата не е реализирана за този параметър.

1. Конфигуриране на FTP сървър с FileZilla
2. Изтегляне сървъра на **FileZilla**.
3. Избиране на порт(препоръчително е да си остане по подразбиране).

A screenshot of a computer

Description automatically generated

1. Задаване на административна парола(по избор).



1. Конфигуриране на групи – „Edit -> Groups -> Add“

A screenshot of a computer

Description automatically generated

1. Създаване на потребител – „Edit -> User -> Add”

A screenshot of a computer

Description automatically generated

1. Активиране на акаунт – „General -> Account Settings“

A screenshot of a computer

Description automatically generated

1. Сравнение между SSH и FTP

|  |  |  |
| --- | --- | --- |
| Разлика | FTP | SSH |
| Цел | изпраща файлове от един хост на друг. | Използва се за прехвърляне на данни през защитен шел канал. |
| Порт | 20,21 | 22 |
| Комуникация | двупосочна комуникация. | SSH терминал за удостоверяване на отдалечено влизане |
| Аутентикация | Потребителско име и парола. | пароли, публичен ключ, GSSAPI, клавиатура или OTP метод. |
| Шифроване | не използва никакъв метод за шифроване. | Използва симетрични, асиметрични и хеширащи алгоритми за криптиране. |
| Преносимост | Той може да не е преносим за среди със защитни стени или подобни приложения за сигурност. | по-преносим от FTP и работи в Unix |
| Функции | копиране, преместване, изтриване и т.н. | използва команди на обвивката за изпълнение на отдалеченото устройство |
| Скорост | По-бърз | По-бавен |
| Сигурност | Няма алгоритъм за сигурност | Висока сигурност |

1. Обобщение

* **FTP** изпраща файлове от един хост на друг, като използва заявки от командния ред заедно с процес на удостоверяване.
* **SSH** е предназначен за комуникация между мрежови устройства в защитен или криптиран режим по незащитен канал.