FTP

SFTP, SCP

Spis treści

FTP – wprowadzenie.

FTP (File Transfer Protocol) – opiera się na modelu klient→serwer.
Istnieją dwa typy usług FTP – uwierzytelnione (standardowe) oraz anonimowe.
Standardowy dostęp do serwera FTP wymaga podania prawidłowej pary – nazwy użytkownika i hasła, natomiast serwer anonimowy pozwala na dostęp każdemu.

FTP nie jest jednak protokołem bezpiecznym.
W dzisiejszych czasach możemy skorzystać z nowych bezpieczniejszych aplikacji, które połączą zalety FTP z bezpiecznym środowiskiem. Pakiet Secure Shell (SSH) posiada takie narzędzia jak **sftp** (secure FTP) czy **scp** (secure copy), które zapewniają większe bezpieczeństwo przesyłanie plików.

Istnieje wiele darmowych serwerów FTP, dedykowanych na platformę Windows i Linux. W Suse Linux możemy skorzystać z dostępnego w repozytoriach systemu pakietu vsftpd (Very Secure FTP).

Typy przesyłanych plików:
ascii – włączamy tryb przesyłania plików jako tekstu ASCII. Należy go używać podczas przesyłania plików tekstowych. Pamiętaj, że pliki html, php itp. są również plikami tekstowymi, tak więc do przetransportowania ich na serwer WWW powinienieś użyć właśnie trybu ASCII.
bin – tryb binarny. Tego trybu używamy podczas transportu plików programów, obrazów, czy spakowanych archiwów. Pliki binarne przesłane w trybie ASCII na serwer stają się bezużyteczne….

Jeżeli na serwerze FTP mamy skonfigurowanego firewall-a należy pamiętać o otwarciu portu 20 (dane), 21 (polecenia). Ponadto pamiętaj, że odpowiednia konfiguracja firewall-a zależy od tego, czy FTP uruchamiamy w trybie aktywnym, czy pasywnym.

Tryb aktywny:
Polecenia: klient > 1024 ------> serwer 21
Dane: klient > 1024 ----------> serwer 20

Tryb pasywny:

Polecenia: klient > 1024 ------> serwer 21

Dane: klient > 1024 ----------> serwer > 1024

Dla trybu pasywnego musisz zatem otworzyć porty wychodzące > 1024, ponieważ przydzielane są losowo. Jeżeli chcemy ograniczyć nr portów do danego zakresu wykonamy to za pomocą następujących dyrektyw znajdujących się w pliku konfiguracyjnym vsftpd:

pasv_min_port=4000

pasv_max_port=4200

pasv_address=ip_naszego_serwera.

Tryb pasywny wykorzystywany jest przez większość przeglądarek internetowych, dlatego aby być uniwersalnym zazwyczaj umożliwiamy obsługę obydwu trybów. Korzystając z powyższych opcji konfiguracyjnych zawężamy zakres otwieranych portów. Za pomocą iptables możemy też „wypuszczać" połączenia wcześniej nawiązane za pomocą modułu **state,** który sprawdza stan połączenia.

Instalacja vsftpd.

Sprawdzamy, czy pakiet nie został już zainstalowany:
Sposób 1: rpm –qa |grep vsftpd
Sposób 2: zypper search vsftpd

Z konsoli wydaj polecenie:
zypper install vsftpd

Uruchomienie, zatrzymanie, restart serwera vsftpd.

/etc/init.d/vsftpd [start | stop | restart | status]

W nowszych wersjach Linux:

systemctl [start | stop | restart | status] vsftpd.service

Dodanie / usunięcie do / z autostartu:

chkconfig [--add | --del] vsftpd

W nowszych wersjach Linux:

systemctl [enlabe | dislabe] vsftpd.service

Główny plik konfiguracyjny demona vsftpd.

Główny plik konfiguracyjny demona vsftpd to: **/etc/vsftpd.conf**
Domyślne ustawienia zostały dostosowane do minimalnej konfiguracji serwera o dostępie anonimowym, oraz trybie uruchomieniowym jako demon.
Po każdej rekonfiguracji w/w pliku należy zrestartować usługę vsftpd !!!

Zmiana domyślnej konfiguracji podczas startu (zmiana domyślnego pliku konfiguracyjnego).

Należy nadmienić, że możliwe jest używanie kilku plików konfiguracyjnych na jednym serwerze ftp i uruchamianie serwera z wybranymi, aktualnie potrzebnymi parametrami z wybranego pliku konfiguracyjnego. Składnia polecenia uruchomienia jest wtedy następująca:

Ścieżka_do_programu_vsftpd nazwa_pliku.conf &

np. /usr/sbin/vsftpd /etc/vsftpd2.conf &

Przy poleceniu wydanym jak powyżej zostanie uruchomiony serwer FTP z opcjami konfiguracyjnymi zamieszczonymi w pliku vsftpd2.conf

Uruchomienie kilku serwerów jednocześnie na jednej maszynie.

Jest również możliwe uruchomienie kilku konfiguracji (kilku serwerów) FTP jednocześnie. Wtedy należy pamiętać o tym, aby w poszczególnej konfiguracji w pliku conf zmienić opcję **listen_port** ! Opcja ta podaje nr portu na jakim nasłuchiwać będzie serwer FTP, chyba, że posiadamy kilka adresów IP (więcej kart sieciowych) to wtedy w pliku konfiguracyjnym zmieniamy opcję **listen_adres** na odpowiedni adres IP na którym będzie nasłuchiwał serwer FTP.

Wybrane opcje konfiguracyjne (dostęp anonymous).

Sekcja Anonymous FTP User Settings zawiera reguły dostępu użytkowników anonimowych. Poniżej przedstawiono opis niektórych z nich:

Aby wyłączyć dostęp anonimowy zmieniamy, lub dopisujemy wartość następującego parametru

anonymous_enable = no

Aby pozwolić na anonimowe pobieranie tylko tych plików, które wszyscy mogą odczytać, należy ustawić wartość parametru

anon_world_readable_only = yes.

Pozwalamy anonimowym użytkownikom wysyłać dane do serwera:

anon_upload_enable = yes

Pozwalamy anonimowym użytkownikom na tworzenie katalogu:

anon_mkdir_write_enable = yes.

Pozwalamy anonimowym użytkownikom zapisywać (oraz edytować i usuwać) pliki w katalogach publicznych :

anon_other_write_enable = yes

Dostęp do serwera dla użytkowników systemowych.

Należy pamiętać, że użytkownik, który ma konto na naszym serwerze po wywołaniu połączenia FTP loguje się automatycznie do swojego katalogu domowego. Więc musimy użytkownikowi taki katalog domowy stworzyć (jeśli system tego nie robi automatycznie). Jeżeli tworzymy ten katalog jako ROOT nie zapominajmy zmienić właściciela na nazwę użytkownika, który powinien być właścicielem tego (swojego) katalogu.

Przyklad konfiguracji /etc/vsftpd.conf dla uzytkownikow systemowych.

```
# Pozwolenie na zapis do wlasnego katalogu
write_enable=YES
# Baner powitalny widziany przez uzytkownika przed zalogowaniem
ftpd_banner="Witam na serwerze FTP."
# Pozwolenie na logowanie uzytkownikom systemowym
local_enable=YES
# Pozwalamy na polaczenia rowniez na porcie 20 - domyslnie tylko 21
connect_from_port_20=YES
# Opcja obowiazkowa - bez niej nie zadziala polaczenie FTP z innej maszyny
pam_service_name=vsftpd
#Uruchomienie serwera FTP w trybie demona poprzez polecenie /usr/sbin/vsftpd &
```

listen=YES

Maska dla tworzonych plikow

local_umask=022

Domyślnie ustawienie local_umask to 077, czyli wysyłane pliki mają następujące uprawnienia: 700. Jak można zauważyć, w takim przypadku tylko właściciel ma dostęp do tych plików. Jeżeli jest to serwer internetowy, po wystawieniu plików html, czy php użytkownicy nie będą ich mogli otwierać w przeglądarce, ponieważ nie mamy do nich uprawnień odczytu…. W serwerach internetowych należy ustawić parametr local_umask na local_umask=022. Takie ustawienie pozwoli na odczytu plików html przez przeglądarkę i w efekcie zaprezentowanie naszej strony.

Wybrane opcje konfiguracyjne dla użytkowników systemowych.

Czas wyrażony w sekundach, definiujący to, ile użytkownik może być w bezczynności (idle) - standardowo (300 s):

idle_session_timeout=300

Opcja odpowiadająca za tzw. banner, czyli komunikat, który będzie wyświetlany podczas każdego połączenia z naszym serwerem, działa ona podobnie jak opcja ftpd_banner, z tym, że tutaj tworzymy osobny plik do takiego bannera i możemy wstawić dłuższy komunikat (domyślnie ustawiamy tu ścieżkę do tego pliku, po uprzednim jego utworzeniu):

banner_file=/usr/local/etc/vsftpd/banner

Definiujemy nazwę pliku, który podobnie jak w opcjach ftpd_banner oraz banner_file będzie wyświetlał komunikat powitalny, po poprawnym zalogowaniu. Plik ten (.message) należy utworzyć w katalogu głównym użytkownika:

message_file=.message

Definiujemy nawy katalogów lub rozszerzenia do plików, które będą widoczne, ale nie będzie możliwości na manipulowanie nimi (zmianę nazwy, pobieranie ich z serwera, kopiowanie itd.). Przykładowo, chcemy aby dany użytkownik nie miał dostępu do katalogu files/ i nie mógł nic w nim zmieniać, a także aby nie mógł nic zrobić z plikami o rozszerzeniach *.mp3 oraz *.avi, rozszerzenia te umieszczamy w klamrach i oddzielamy je przecinkami:

deny_file={*.mp3,files/,*.avi}

Opcja ta ukrywa nazwy katalogów / plików. Podobnie jak powyżej nazwy rozszerzeń plików lub nazwy katalogów umieszczamy pomiędzy dwoma klamrami i oddzielamy je przecinkami, przykładowo chcemy ukryć przed użytkownikiem wszystkie pliki, które będą miały rozszerzenie *.doc:

hide_file={*.doc}

Poniższe dwie opcje odpowiadają za listę użytkowników, którzy nie będą mieli pozwolenia na logowanie do serwera, po wpisaniu nazwy użytkownika, który będzie znajdował się na tej liście podczas logowania pojawi się komunikat 'Permission denied.'. Nazwy tych użytkowników możemy dopisywać poprzez użycie komendy echo, lub edytując ten plik i wpisując loginy, każdy w osobnej linii:

userlist_enable=YES

userlist_file=/usr/local/etc/vsftpd/userlist

W opcji tej ustawiamy adres ip, na którym vsftpd ma nasłuchiwać. Przydatna jeśli chcemy uruchomić większą liczbę serwerów na jednym komputerze. Oczywiście jeśli mamy odpowiednią ilość adresów ip.

listen_address=111.222.333.444

Ciekawą opcją jest user_config_dir, która pozwala przyporządkować dowolnemu użytkownikowi w systemie konkretne opcje. Mogą to być np. takie opcje jak listen_address, banner_file, max_per_ip, max_clients, xferlog_file, vsftpd_log_file, itp. Definiujemy więc ścieżkę do takiego katalogu, oraz tworzymy go w systemie. Po zdefiniowaniu tej opcji w pliku konfiguracyjnym, vsftpd będzie automatycznie szukał pliku, który nosi taką samą nazwę jak użytkownik systemowy. Na przykład dla użytkownika `pkania` takim plikiem będzie /usr/local/etc/vsftpd/user_conf/pkania, w tym właśnie pliku będziemy ustawiać konkretne opcje dla tego użytkownika.

user_config_dir=/usr/local/etc/vsftpd/user_conf/

Graficzny klient FTP.

Istnieje wiele graficznych klientów FTP. Dla systemu Windows to np. Total Commander.
W systemie Linux gFTP, czy KBear.

Pamiętajmy również, że każda przeglądarka internetowa może posłużyć jako klient ftp. W pasku adresu danej przeglądarki należy wpisać zamiast http://adres_serwera_www następującą frazę (zaczynając od słowa ftp) : ftp://adres_serwera_ftp

Tekstowy klient FTP.

Polecenia uruchamiania klienta i połączenia:
ftp – uruchamia klienta FTP.
open – nawiązuje połączenie z serwerem.
close – zamyka połączenie, lecz nie wychodzi z klienta ftp.
bye, quit – zamyka połączenie, wychodzi z klienta ftp.

Przykłady połączenia (dwa sposoby):
1) ftp ip_lub_nazwa_serwera
2) W wierszu zachęty wydajemy polecenie: ftp i enter,
Następnie : open ip_lub_nazwa_serwera

Ogólne zasady pracy / wydawanie komend :
Jeżeli na komputerze lokalnym (kliencie ftp) chcemy zmienić katalog (np. na temp) użyjemy polecenia :
lcd temp
Aby wykonać polecenie powłoki, nie wychodząc z sesji ftp użyjemy znaku ! (wykrzyknika) przed poleceniem powłoki.
Przydatnym poleceniem jest polecenie „**syst**”, po wydaniu którego pokaże się informacja o systemie operacyjnym serwera i wersji demona FTP.

Polecenia wykonujące działania na plikach / katalogach:
get – pobieranie pojedynczego pliku z serwera,
put, send – wysłanie pojedynczego pliku na serwer,
mget – pobranie wielu plików na raz z serwera, można używać masek typu *.*
mput – wysyłanie wielu plików na serwer równocześnie,
hash – włącza wyświetlanie przez klienta FTP paska postępu transmisji składającego się z ciągu znaków ######

SFTP.

Sftp można użyć wtedy, gdy na zdalnym PC zainstalowany jest OpenSSH. W OpenSSH dostęp przez sftp jest domyślnie włączony.

Aby użyć sftp w trybie interaktywnym, należy w poleceniu podać adres IP lub nazwę zdalnego hosta:

sftp <adres IP>

lub

sftp docs.protek.edu

Po podłączeniu do zdalnego hosta używamy poleceń tekstowych ftp jak : get, put itd..

Domyślnie zdalny komputer rozpoznaje zalogowanego użytkownika. Aby użyć w zdalnym komputerze innej nazwy użytkownika, należy użyć następującej składni:

sftp pkania@docs.protek.edu

Powyższe polecenia nawiąże połączenie z serwerem docs.protek.edu jako użytkownik pkania.

Najczęściej demon sshd używa portu 22. Jeżeli administrator zmienił nr portu w wywołaniu sftp musimy ten port podać:

sftp -o –P 2222 pkania@docs.protek.edu

SCP.

W sytuacji, gdy sam transfer pliku też powinien być bezpieczny należy posłużyć się programem **scp**, zamiast sftp. Transmisja plików za pomocą scp jest powolna (dane są szyfrowane), ale za to mamy bezpieczny transport plików przez sieć.

opcje SCP:

- `-r` – kopiuje rekursywnie (z podkatalogami) podany katalog,
- `-P port` – używa innego portu niż standardowy 22 (oczywiście używamy tej opcji w przypadku, gdy zdalny serwer SSH nasłuchuje na niestandardowym porcie).

Kopiowanie pliku :

scp plik_lokalny serwer:/katalog_zdalny/

np.: scp firewall.sh 192.168.1.101:/root

Pobieranie pliku z serwera do hosta lokalnego (polecenie wydajemy na hoście lokalnym):

scp –r serwer:/katalog_zdalny/plik /katalog_lokalny/

np.: scp -r 192.168.1.101:/root/start_firewall.sh /root

SSH, SCP bez hasła.

Generujemy klucze:

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/użytkownik/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/użytkownik/.ssh/id_rsa.
Your public key has been saved in /home/użytkownik/.ssh/id_rsa.pub.
The key fingerprint is:
xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx
```

Przy pytaniu o hasło należy wcisnąć *ENTER* bez podania hasła. Zostanie utworzony klucz bez hasła. Po wykonaniu tego polecenia, wygenerowaliśmy dwa klucze. Klucz prywatny, zapisany w pliku */home/użytkownik/.ssh/id_rsa* (nie udostępniamy go nikomu). Drugi klucz, publiczny, znajduje się w pliku */home/użytkownik/.ssh/id_rsa.pub* (ten klucz udostępniamy).

Dodajemy wpis o naszym kluczu publicznym do pliku *authorized_keys*, który znajduje się w katalogu ~/.ssh na serwerze zdalnym. Wydaj polecenia:

```
scp /home/użytkownik/.ssh/id_rsa.pub użytkownik@zdalny_serwer:~/
ssh użytkownik@zdalny_serwer
cat ~/id_rsa.pub >> ~/.ssh/authorized_keys
```

Po tej operacji wszelkie akcje wykonywane na zdalnym serwerze za pośrednictwem SSH nie będą wymagały podawania hasła np.: logowanie, przesyłanie plików przez scp itp.

WinSCP.

Jeżeli w systemie Windows chcemy korzystać z bezpiecznego przesyłania plików przez ssh możemy wykorzystać program **WinSCP**, Narzędzie to działa zarówno w trybie graficznym, jak i tekstowym.

Przykład wykorzystania WinSCP do synchronizacji katalogów Linux / Windows.

1) Tworzymy plik o nazwie np. synchronizacja.bat (najlepiej w domyślnej lokalizacji WinScp):
   cd C:\Program Files\WinSCP

winscp.exe /console /script=skrypt.txt

if errorlevel 1 goto error

echo "Kopiowanie poprawne" >> C:\Program Files\WinSCP\log_KANIA.txt

echo "----------">> C:\Program Files\WinSCP\log_KANIA.txt

:error

echo "błąd kopiowania" >> C:\Program Files\WinSCP\log_KANIA.txt

echo "----------">> C:\Program Files\WinSCP\log_KANIA.txt

2) Tworzymy plik o nazwie na którą wskazuje opcja /script, tu skrypt.txt (najlepiej w domyślnej lokalizacji WinScp):

Automatically answer all prompts negatively not to stall

the script on errors

option batch on

Disable overwrite confirmations that conflict with the previous

option confirm off

Connect using a password

open user:password@example.com

Connect

open root:haslo_usera@ip_serwera_zdalnego

Change remote directory

cd /srv/przyklad/archiwum/

Force binary mode transfer

option transfer binary

#Synchronizacja katalogów opcja local mówi, ze synchronizowany katalog

#to katalog lokalny, jezeli chcialbym synchronizowac katalog zdalny to opcja remote

#obydwa "lustro" both np: synchronize both d:\www /home/martin/public_html

synchronize local c:\kopia\ /srv/przyklad/archiwum/

Disconnect

close

exit

3) Dodajemy zadanie do harmonogramu Windows, które uruchamia automatycznie synchronizację w/w katalogów o danej porze (jednorazowo lub cyklicznie).

Dokładny opis poszczególnych opcji WinSCP znajdziesz pod poniższym adresem:
http://winscp.net/eng/docs/scripting#commands