Serwer SFTP

Typ: procedura **Dotyczy: OpenSSH na RHEL/Rocky i Debian/Ubuntu** **Czas: ~30 min**

Efekt: nowy użytkownik zamknięty w swoim katalogu (chroot), logujący się wyłącznie po SFTP kluczem SSH, bez dostępu do powłoki i bez możliwości wyjścia poza własny katalog.

Skróty użyte na tej stronie (6)

Skrót	Co znaczy
FTP	File Transfer Protocol: stary protokół przesyłania plików, bez szyfrowania
FTPS	FTP Secure: protokół FTP z szyfrowaniem TLS
RHEL	Red Hat Enterprise Linux: komercyjna dystrybucja Linuksa
SFTP	SSH File Transfer Protocol: bezpieczne przesyłanie plików przez SSH
SSH	Secure Shell: szyfrowane zdalne logowanie do serwera
TLS	Transport Layer Security: szyfrowanie połączeń, np. w HTTPS

Czym jest SFTP

Pojęcie: SFTP (SSH File Transfer Protocol). Protokół przesyłania plików działający w całości wewnątrz jednej sesji SSH, na tym samym porcie 22 co zwykłe logowanie terminalowe — w odróżnieniu od FTP, nie otwiera osobnych, dodatkowych połączeń na inne porty do przesyłania danych, więc jest prostszy do przepuszczenia przez firewall i domyślnie w pełni szyfrowany, bo dziedziczy szyfrowanie z samego SSH.

Pojęcie: FTPS. FTP rozszerzone o TLS — inny protokół niż SFTP, mimo podobnej nazwy i podobnego zastosowania. FTPS korzysta z oddzielnych połączeń kontrolnych i danych (tak jak zwykłe FTP), więc wymaga otwarcia szerszego zakresu portów na firewallu, a jego bezpieczeństwo zależy od poprawnej konfiguracji certyfikatu TLS, zupełnie osobno od kluczy SSH. Konta konfigurowane poniższą procedurą korzystają wyłącznie z SFTP — nie z FTPS.

Zanim zaczniemy

> **Z ŻYCIA WZIĘTE**
>
> O 8:15 dział księgowości zgłasza Tomkowi, że kontrahent z zewnątrz od rana nie może wgrać pliku na `app01`, bo dostaje `Connection closed` zaraz po podaniu hasła. Tomek sprawdza `ls -ld` na katalogu chroot i widzi, że ktoś wieczorem poprzedniego dnia odpalił na nim `chmod 775` przy porządkowaniu uprawnień, bo katalog wyglądał „za bardzo zamknięty". Po przywróceniu `root:root` i `755` logowanie od razu wraca do działania.

Wymaganie	Jak sprawdzić
Serwer OpenSSH zainstalowany i działający	`systemctl status sshd.service`
Katalog nadrzędny dla kont SFTP istnieje na dysku (np. `/ftp`)	`ls -ld /ftp`
Grupa systemowa dla kont SFTP istnieje	`getent group sftpusers`
Klucz publiczny użytkownika dostarczony z wyprzedzeniem	plik `.pub` od użytkownika, nie generujemy go za niego

Uważaj: wymagania własnościowe katalogu chroot są bardzo sztywne i to jest najczęstszy powód, dla którego logowanie nie działa mimo poprawnej reszty konfiguracji — katalog wskazany w `ChrootDirectory` oraz każdy katalog nadrzędny na ścieżce do niego musi należeć do roota (`root:root`) i nie może mieć prawa zapisu dla grupy ani dla innych. OpenSSH odmawia użycia chroot, jeśli choć jeden katalog na tej ścieżce jest zapisywalny przez kogokolwiek poza rootem — traktuje to jako lukę bezpieczeństwa, bo użytkownik zamknięty w chroot mógłby inaczej podmienić zawartość katalogu, z którego korzysta sam proces SSH.

Przebieg

1. Utworzenie grupy dla kont SFTP

```
sudo groupadd sftpusers
```

Jedna wspólna grupa systemowa, do której będą należeć wszyscy użytkownicy objęci regułami chroot z kroku 5 — blok `Match Group` w `sshd_config` odwołuje się do niej po nazwie, więc dodanie kolejnego użytkownika do tej grupy automatycznie obejmuje go tymi samymi regułami, bez dodatkowej zmiany w `sshd_config`.

Weryfikujemy:

```
getent group sftpusers
```

Przykład:

```
sftpusers:x:1050:
```

2. Utworzenie konta użytkownika bez powłoki

```
sudo useradd -d /ftp/nazwa_usera -g sftpusers -s /usr/sbin/nologin nazwa_usera
```

> (gdzie `nazwa_usera` to nazwa konta systemowego tworzonego dla klienta albo integracji korzystającej z SFTP)

Powłoka `/usr/sbin/nologin` uniemożliwia temu kontu zwykłe logowanie terminalowe przez SSH — w połączeniu z `ForceCommand internal-sftp` z kroku 5 i tak wymusilibyśmy sesję SFTP, ale ustawienie `nologin` już na poziomie samego konta jest dodatkową warstwą zabezpieczenia, niezależną od konfiguracji SSH.

3. Ustawienie hasła konta (opcjonalnie, przy braku logowania kluczem)

```
sudo passwd nazwa_usera
```

Wymagane tylko wtedy, gdy to konto ma się logować hasłem zamiast kluczem — w docelowej konfiguracji z krokiem 7 (klucz zamiast hasła) ten krok pomijamy albo blokujemy hasło komendą `passwd -l nazwa_usera` po dodaniu klucza.

Przykład:

```
New password:
Retype new password:
passwd: password updated successfully
```

4. Utworzenie struktury katalogu chroot

```
sudo mkdir -p /ftp/nazwa_usera/dane
sudo chown root:root /ftp/nazwa_usera
sudo chmod 755 /ftp/nazwa_usera
sudo chown nazwa_usera:sftpusers /ftp/nazwa_usera/dane
```

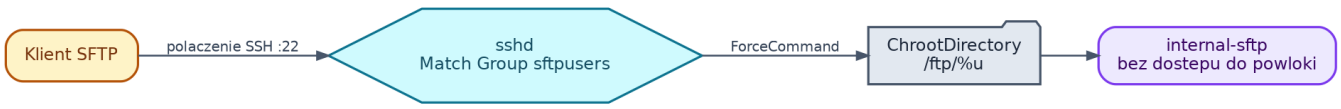
Katalog główny użytkownika (`/ftp/nazwa_usera`) musi zostać własnością roota, zgodnie z wymaganiem z sekcji „Zanim zaczniemy" — dopiero podkatalog wewnątrz niego (`dane`) należy do samego użytkownika i to w nim faktycznie ląduje przesyłane przez niego pliki. Próba zapisu bezpośrednio do katalogu głównego chroot kończy się odmową uprawnień, co jest zamierzonym zachowaniem, nie usterką.

5. Dodanie bloku Match Group w sshd_config

```
sudo tee -a /etc/ssh/sshd_config <<'EOF'

Match Group sftpusers
    ChrootDirectory /ftp/%u
    ForceCommand internal-sftp
    AllowTcpForwarding no
    X11Forwarding no
EOF
```

> **Pojęcie: Match Group.** Blok w `sshd_config`, którego dyrektywy obowiązują tylko dla połączeń użytkowników należących do wskazanej grupy — wszystko poza tym blokiem działa dla nich tak samo jak dla reszty serwera, ale wewnątrz bloku nadpisujemy kluczowe ustawienia specyficzne dla kont SFTP.



Połączenie SFTP przechodzi kolejno przez sshd, katalog chroot wskazany przez ForceCommand i wbudowaną implementację internal-sftp (kliknięcie otwiera pełny rozmiar)

`%u` w ścieżce `ChrootDirectory` rozwija się automatycznie do nazwy loginu łączącego się użytkownika, więc jeden wpis w konfiguracji obsługuje dowolną liczbę kont bez dopisywania

osobnej reguły na każde z nich.

> **Pojęcie: ForceCommand internal-sftp.** Niezależnie od tego, jaką komendę użytkownik próbowałby wywołać przy połączeniu, serwer ignoruje ją i zawsze uruchamia wbudowaną w OpenSSH implementację SFTP (`internal-sftp`), działającą w pamięci bez odwołania do zewnętrznego binarnego pliku wykonywalnego wewnątrz chroot — dzięki temu nie musimy w ogóle kopiować żadnych binariów do środowiska chroot, jak byłoby to konieczne przy klasycznym `/usr/lib/openssh/sftp-server`.

`AllowTcpForwarding no` blokuje wykorzystanie tego konta jako tunelu SSH do innych hostów w sieci — bez tego ograniczenia konto ograniczone do SFTP nadal mogłoby posłużyć jako punkt przeskoku sieciowego, mimo braku dostępu do powłoki.

6. Sprawdzenie składni i przeładowanie sshd

```
sudo sshd -t
```

Sprawdzamy składnię konfiguracji przed przeładowaniem — błąd w `sshd_config` sprawdzony dopiero po restarcie usługi potrafi odciąć wszystkie sesje SSH do serwera, łącznie z sesją administratora.

```
sudo systemctl restart sshd.service
```

7. Konfiguracja logowania kluczem zamiast hasłem

```
sudo mkdir -p /ftp/nazwa_usera/dane/.ssh
```

Katalog `.ssh` umieszczamy wewnątrz podkatalogu należącego do użytkownika (`dane`), nie w katalogu głównym chroot — z tych samych powodów własnościowych co w kroku 4.

Wgrywamy dostarczony wcześniej klucz publiczny użytkownika do pliku `authorized_keys`:

```
sudo tee /ftp/nazwa_usera/dane/.ssh/authorized_keys < SciezkaDoPliku
```

Ustawiamy wymagane, restrykcyjne uprawnienia:

```
sudo chown -R nazwa_usera:sftpusers /ftp/nazwa_usera/dane/.ssh
sudo chmod 700 /ftp/nazwa_usera/dane/.ssh
sudo chmod 600 /ftp/nazwa_usera/dane/.ssh/authorized_keys
```

Uważaj: OpenSSH odmawia użycia pliku `authorized_keys`, jeśli katalog `.ssh` albo sam plik mają zbyt szerokie uprawnienia (np. zapis dla grupy) — te same zasady bezpieczeństwa, co przy zwykłym logowaniu SSH kluczem, obowiązują też wewnątrz chroot.

Wskazujemy niestandardową ścieżkę pliku kluczy w bloku `Match Group`, jeśli różni się od domyślnej lokalizacji względem katalogu domowego:

```
sudo sed -i '/Match Group sftpusers/a\    AuthorizedKeysFile
/ftp/%u/dane/.ssh/authorized_keys' /etc/ssh/sshd_config
sudo sshd -t && sudo systemctl restart sshd.service
```

Weryfikacja

Sprawdzenie logowania kluczem

```
sftp -i SciezkaDoKlucza -P 22 nazwa_usera@NazwaSerwera
```

Przykład udanego połączenia:

```
Connected to NazwaSerwera.
sftp> pwd
Remote working directory: /
sftp> ls
dane
sftp> cd dane
sftp> put plik_testowy.txt
Uploading plik_testowy.txt to /dane/plik_testowy.txt
plik_testowy.txt                              100%   12    0.1KB/s   00:00
```

Katalog widziany po zalogowaniu to `/` — to jest właściwe zachowanie chroot: użytkownik widzi katalog `/ftp/nazwa_usera` jako swój własny korzeń systemu plików i nie ma żadnej możliwości wyjścia poza niego, nawet komendą `cd ..`.

Sprawdzenie uprawnień katalogu chroot

```
ls -ld /ftp/nazwa_usera
```

Przykład poprawnego wyniku:

```
drwxr-xr-x. 3 root root 4096 Sep 24 10:15 /ftp/nazwa_usera
```

Właściciel `root:root` i brak prawa zapisu dla grupy ani innych (`w` obecne tylko w pierwszej trójce uprawnień) potwierdzają zgodność z wymaganiem z sekcji „Zanim zaczniemy".

Wycofanie zmian

Usunięcie konta i jego danych, gdy dostęp SFTP nie jest już potrzebny:

```
sudo userdel nazwa_usera
```

```
sudo rm -rf /ftp/nazwa_usera
```

Usunięcie bloku `Match Group` z `sshd_config` ma sens dopiero, gdy usuwamy ostatnie konto należące do grupy `sftpusers` — dopóki istnieje choć jedno takie konto, blok zostaje.

Gdy coś nie zadziała

Objaw	Przyczyna	Naprawa
`Connection closed` zaraz po próbie logowania	katalog `ChrootDirectory` albo katalog nadrzędny ma złego właściciela albo zbyt szerokie uprawnienia	`ls -ld` na każdym katalogu ścieżki aż do `/`, poprawa na `root:root` i `755` lub węziej
`Permission denied (publickey)` mimo poprawnego klucza	zły właściciel albo uprawnienia `.ssh`/`authorized_keys`, albo klucz w niewłaściwej lokalizacji	sprawdzenie `chmod 700`/`600` i ścieżki wskazanej w `AuthorizedKeysFile`

Objaw	Przyczyna	Naprawa
Użytkownik widzi więcej niż powinien, w tym pliki systemowe	brak albo błędny blok `Match Group`, `ChrootDirectory` nie zadziałał	`sudo sshd -T

Objaw	Przyczyna	Naprawa
Logowanie działa, ale użytkownik trafia do powłoki zamiast sesji SFTP	brak `ForceCommand internal-sftp` w bloku `Match Group`	dodanie dyrektywy i przeładowanie `sshd`
`sshd -t` zgłasza błąd składni po edycji	literówka w nazwie dyrektywy albo zły poziom wcięcia bloku `Match`	poprawa pliku, ponowne `sshd -t` przed jakimkolwiek restartem usługi

Powiązane: Apache Solr: utrzymanie · Certyfikaty SSL: wystawienie, konwersja, weryfikacja

Źródła: `man 5 sshd_config` · `man 8 sshd` · dokumentacja OpenSSH, rozdział o `ChrootDirectory` i `internal-sftp`

Revision #36
Created 2026-09-25 07:40:25 UTC by Tomasz Konieczny
Updated 2026-10-26 14:14:23 UTC by Tomasz Konieczny