SSH Keys
Configurere Autentificare fără Parolă

Timp lucru 45 minute

1. Introducere

SSH Keys oferă o metodă de autentificare mai sigură și mai convenabilă decât parolele. Odată configurată, poți conecta la server fără să introduci parola de fiecare dată.

Avantaje SSH Keys:

- Mai sigur decât parolele (criptare 2048+ biți)
- Nu mai introduci parola la fiecare conectare
- Poți dezactiva complet autentificarea cu parolă
- Necesar pentru automatizări (scripturi, CI/CD)

2. Cum Funcționează SSH Keys?

SSH folosește criptografie asimetrică cu o pereche de chei:

Cheia PRIVATĂ (id_rsa):

• Rămâne pe computerul TĂU (client)
• NU o partajezi NICIODATĂ cu nimeni!
• Este ca o parolă foarte sigură
• Locație: ~/.ssh/id_rsa

Cheia PUBLICĂ (id_rsa.pub):

• O copiezi pe SERVER
• E sigur să o distribui (de aici numele "publică")
• Se adaugă în fișierul ~/.ssh/authorized_keys de pe server
• Locație pe client: ~/.ssh/id_rsa.pub

Diagrama Proces

```
CLIENT                          SERVER
  |
  |-- Generează pereche chei
  |    ├── id_rsa (PRIVATĂ) → Păstrează LOCAL
  |    └── id_rsa.pub (PUBLICĂ) → Copiază pe SERVER
  |
  |-- ssh-copy-id user@server
  |                               |
  |                               |-- Primește cheia publică
  |                               |-- Adaugă în ~/.ssh/authorized_keys
  |
  |-- ssh user@server (fără parolă!)
  |                               |
  |-- Trimite proof cu cheia privată
  |                               |-- Verifică cu cheia publică
  |                               |-- ✓ Acces permis!
  |<---- Conectat fără parolă ---|
```

!!! *Cheia PRIVATĂ nu părăsește NICIODATĂ computerul tău! Doar cheia PUBLICĂ merge pe server.*

3. Generarea Perechii de Chei SSH

3.1 Verificare Chei Existente

Înainte de a genera chei noi, verifică dacă există deja:

```
ls -la ~/.ssh/

# Caută fișiere:
id_rsa        # Cheie privată
id_rsa.pub    # Cheie publică
```

Dacă există deja și vrei să le folosești, sari la pasul 4 (copierea pe server).

3.2 Generare Chei Noi

Comandă pentru generare:

```
ssh-keygen -t rsa -b 4096 -C "email@example.com"
```

Explicații parametri:

• -t rsa = tip algoritm (RSA recomandat)
• -b 4096 = dimensiune cheie (4096 biți = foarte sigur)
• -C "comentariu" = identificare cheie (folosește email-ul)

3.3 Proces Interactiv

După rularea comenzii, vei fi întrebat:

1. Unde să salveze cheia:

```
Enter file in which to save the key (/home/user/.ssh/id_rsa):
```

→ Apasă ENTER pentru locația default (recomandat)

2. Passphrase (opțional):

```
Enter passphrase (empty for no passphrase):
```

→ Poți lăsa gol (ENTER) pentru no passphrase
→ SAU setezi o parolă extra pentru cheia privată (mai sigur)

!!! *Passphrase = parolă pentru cheia privată. Dacă setezi, vei introduce passphrase-ul în loc de parola SSH (tot mai convenabil). Dacă lași gol, nu introduci nimic.*

3. Confirmare passphrase:

```
Enter same passphrase again:
```

→ Re-introdu passphrase-ul (sau ENTER dacă l-ai lăsat gol)

3.4 Rezultat

După finalizare, vei vedea:

```
Your identification has been saved in /home/user/.ssh/id_rsa
Your public key has been saved in /home/user/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:abc123xyz... email@example.com
The key's randomart image is:
+---[RSA 4096]----+
|  .o.+           |
|   .= o          |
+----[SHA256]-----+
```

Verifică cheile create:

```
ls -la ~/.ssh/

# Vei vedea:
id_rsa       # Cheie PRIVATĂ (600 permisiuni)
id_rsa.pub   # Cheie PUBLICĂ (644 permisiuni)
```

Vizualizează cheia publică:

```
cat ~/.ssh/id_rsa.pub

# Output (exemplu):
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQ... email@example.com
```

4. Copierea Cheii Publice pe Server

Acum trebuie să copiezi cheia PUBLICĂ (id_rsa.pub) pe server. Există două metode: automată (recomandată) și manuală.

4.1 Metoda 1: ssh-copy-id (RECOMANDATĂ)

Cea mai simplă metodă:

```
ssh-copy-id user@IP_SERVER

# Exemplu:
ssh-copy-id admin@192.168.1.100
```

Procesul:
1. Te va cere parola SSH (ultima dată!)
2. Copiază automat cheia publică în ~/.ssh/authorized_keys pe server
3. Setează automat permisiunile corecte

Output așteptat:

```
/usr/bin/ssh-copy-id: INFO: attempting to log in...
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed
user@192.168.1.100's password: [introduci parola]

Number of key(s) added: 1

Now try logging into the machine with:
   ssh user@192.168.1.100
```

!!! *ssh-copy-id e disponibil pe Linux/Mac. Pe Windows, folosește metoda manuală sau instalează OpenSSH.*

4.2 Metoda 2: Manual (Alternativă)

Dacă ssh-copy-id nu e disponibil:

Pasul 1: Copiază conținutul cheii publice

```
cat ~/.ssh/id_rsa.pub

# Selectează și copiază TOT textul (ssh-rsa AAA...)
```

Pasul 2: Conectează la server cu parolă

```
ssh user@192.168.1.100
# Introdu parola
```

Pasul 3: Creează directorul .ssh (dacă nu există)

```
mkdir -p ~/.ssh
chmod 700 ~/.ssh
```

Pasul 4: Adaugă cheia în authorized_keys

```
nano ~/.ssh/authorized_keys

# Lipește cheia publică (ssh-rsa AAA...)
# Salvează: Ctrl+O, Enter, Ctrl+X
```

Pasul 5: Setează permisiunile corecte

```
chmod 600 ~/.ssh/authorized_keys
chmod 700 ~/.ssh
```

Pasul 6: Deconectează

```
exit
```

5. Testare Autentificare cu Cheie

Acum poți testa autentificarea fără parolă:

```
ssh user@192.168.1.100

# Ar trebui să te conectezi FĂRĂ să introducă parola!
```

Dacă funcționează:
✓ Te conectezi imediat fără parolă
✓ Vei vedea prompt-ul serverului direct
Dacă NU funcționează (încă cere parolă):
Vezi secțiunea Troubleshooting mai jos

6. Securizare Suplimentară (Opțional)

6.1 Dezactivează Autentificarea cu Parolă

După ce SSH keys funcționează, poți dezactiva complet parolele (mai sigur):
!!! *Fă asta DOAR după ce ai confirmat că SSH keys funcționează! Altfel te blochezi!*

```
# Pe SERVER:
sudo nano /etc/ssh/sshd_config

# Modifică/Adaugă:
PasswordAuthentication no
ChallengeResponseAuthentication no
UsePAM no

# Salvează și restart SSH
sudo systemctl restart sshd
```

Testează din alt terminal ÎNAINTE să închizi sesiunea curentă!

6.2 Permisiuni Corecte (Important!)

Permisiunile trebuie setate corect, altfel SSH va refuza să folosească cheile:
Pe CLIENT:

```
chmod 700 ~/.ssh
chmod 600 ~/.ssh/id_rsa
chmod 644 ~/.ssh/id_rsa.pub
```

Pe SERVER:

```
chmod 700 ~/.ssh
chmod 600 ~/.ssh/authorized_keys
```

!!! *SSH este foarte strict cu permisiunile! Dacă sunt prea permisive, va ignora cheile.*

7. Troubleshooting - Probleme Comune

7.1 Încă Cere Parola

Cauze posibile:

1. Permisiuni greșite

```
# Pe server, verifică:
ls -la ~/.ssh/

# Ar trebui:
drwx------ .ssh/
-rw------- authorized_keys

# Corectează:
chmod 700 ~/.ssh
chmod 600 ~/.ssh/authorized_keys
```

2. Cheia publică nu e pe server

```
# Verifică pe server:
cat ~/.ssh/authorized_keys

# Ar trebui să vezi cheia ta (ssh-rsa AAA...)
```

3. SELinux blochează (CentOS/RHEL)

```
# Pe server:
restorecon -R -v ~/.ssh
```

4. Config SSH pe server blochează

```
# Verifică pe server:
sudo grep -i "PubkeyAuthentication" /etc/ssh/sshd_config

# Ar trebui:
PubkeyAuthentication yes

# Dacă e "no", schimbă în "yes" și restart:
sudo systemctl restart sshd
```

7.2 Debug Verbose

Pentru a vedea exact ce se întâmplă:

```
ssh -vvv user@192.168.1.100

# -vvv = foarte verbose (detalii complete)
# Caută linii cu:
#   "Offering public key"
#   "Server accepts key"
#   "Authentication succeeded"
```

7.3 "Permission denied (publickey)"

Înseamnă că serverul nu acceptă cheia ta.

```
# Verifică pe server log-urile:
sudo tail -f /var/log/auth.log     # Ubuntu/Debian
sudo tail -f /var/log/secure       # CentOS/RHEL

# În timpul conectării, vei vedea de ce e respinsă
```

8. Lucrul cu Multiple Chei SSH

Dacă ai mai multe servere sau chei diferite:

8.1 Generare Chei Multiple

```
# Cheie pentru server1
ssh-keygen -t rsa -b 4096 -f ~/.ssh/id_rsa_server1

# Cheie pentru server2
ssh-keygen -t rsa -b 4096 -f ~/.ssh/id_rsa_server2
```

8.2 Config SSH pe Client

Creează fișier de configurare:

```
nano ~/.ssh/config

# Adaugă:
Host server1
    HostName 192.168.1.100
    User admin
    IdentityFile ~/.ssh/id_rsa_server1

Host server2
    HostName 192.168.1.101
    User root
    IdentityFile ~/.ssh/id_rsa_server2

# Salvează
```

Acum poți conecta simplu:

```
ssh server1  # Folosește automat id_rsa_server1
ssh server2  # Folosește automat id_rsa_server2
```

9. Rezumat - Pași Rapidi

9.1 Setup Complet în 3 Comenzi

```
# 1. Generează chei (pe CLIENT)
ssh-keygen -t rsa -b 4096 -C "email@example.com"
# → Apasă ENTER de 3 ori

# 2. Copiază pe server (pe CLIENT)
ssh-copy-id user@192.168.1.100
# → Introdu parola

# 3. Testează (pe CLIENT)
ssh user@192.168.1.100
# → Ar trebui să conectezi FĂRĂ parolă!
```

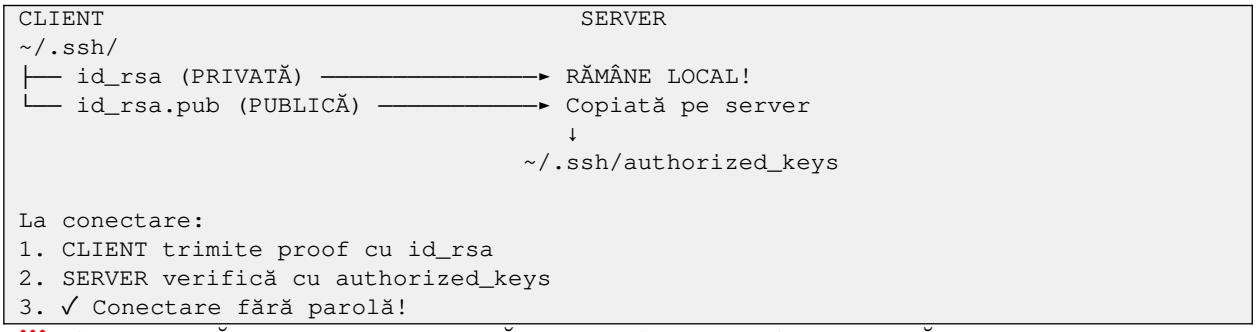
9.2 Checklist Verificare

- ☐ Chei generate (id_rsa și id_rsa.pub)
- ☐ Cheia publică copiată pe server
- ☐ Permisiuni corecte (700 pentru .ssh/, 600 pentru chei)
- ☐ PubkeyAuthentication yes pe server
- ☐ Conectare fără parolă funcționează
- ☐ (Opțional) PasswordAuthentication no pe server

9.3 Comenzi Utile

```
# Generare chei
ssh-keygen -t rsa -b 4096

# Copiază pe server
ssh-copy-id user@host

# Conectare
ssh user@host

# Debug
ssh -vvv user@host

# Verifică permisiuni
ls -la ~/.ssh/

# Vizualizează cheie publică
cat ~/.ssh/id_rsa.pub

# Verifică authorized_keys pe server
cat ~/.ssh/authorized_keys
```

9.4 Diagramă Finală



!!! *Cheia PRIVATĂ nu părăseşte NICIODATĂ computerul tău! Doar cheia PUBLICĂ merge pe server.*