SSH KEY-BASED AUTHENTICATION

Secure Access to Remote Servers

Prasad Rajendra Pansare

AGENDA

- Introduction to SSH Key-Based Authentication
- Understanding Public and Private Keys
- How Key-Based Authentication Works
- Generating SSH Keys
- Adding Public Key to Remote Server
- Demo
- Conclusion

INTRODUCTION

What is SSH Key-Based Authentication?

- A more secure alternative to password-based authentication
- Uses public and private key pairs for verification

- Benefits
- Eliminates the need for remembering complex passwords
- Provides stronger security against brute-force attacks
- Streamlines the authentication process

UNDERSTANDING PUBLIC AND PRIVATE KEYS

- Key Pair Generation
  - Using tools like ssh-keygen
  - Consists of a public key and a private key
- Public Key Distribution
  - Shared with the remote server
- Private Key Security
  - Kept strictly confidential

HOW KEY-BASED AUTHENTICATION WORKS

1 Client generates a key pair.

2 Client sends the public key to the server.

3 Server stores the public key.

4 Client initiates an SSH connection.

5 Server generates a random challenge.

6 Client uses the private key to sign the challenge.

7 Server verifies the signature using the public key .

8 If the signature is valid, authentication is successful.

GENERATING SSH KEYS

- Using ssh-keygen
  - Options:
    - -t: Key type (e.g., RSA, DSA, ECDSA)
    - -b: Key bit length
    - -C: Comment for identification
- Storing Keys
  - Default location: ~/.ssh/id_rsa (private key)
  - ~/.ssh/id_rsa.pub (public key)

ADDING PUBLIC KEY TO REMOTE SERVER

- Authorized Keys File
  - Typically located at ~/.ssh/authorized_keys on the server
- Methods
  - Direct editing: Manually append the public key
  - ssh-copy-id: Convenient command for copying the public key

DEMO

```
root@prasad:~# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:pw1zWrtIqQpcK4ymxVaiqONCOUD7Uzu+5UdUOCIlObO root@prasad
The key's randomart image is:
+---[RSA 3072]----+
|..    .+.  .     |
| ..   +.o o .    |
| .o  .o o o      |
| .... .E .       |
|  oooo  S +      |
|.B +o..  % .     |
|=.O .. .* o      |
|*o o  +o o .     |
|=o  .o..o .      |
+----[SHA256]-----+
```

Ssh-keygen generate

Public key save in encrypted format

Cat /root/.ssh/id_rsa.pub

```
root@prasad:~# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCvwTwHzIvCsszQOrNo2cPNN1/AoW55IivVOI29uCS/A17m9KwCOYJ9vtkjI/65
VmT17QmwTthOpGxpsDS/hkARthb/fkjT2FfxcEAvhhqz7sl9/3yxOqLXBmNxGsPCL4/+eJNe7OI+kwoNfVB/pPpRRYRkrDrTdFCm
xKJuJFsNrY+LbE4JOYKBf6HH9XPOH8ymkz8BI521dXBkd+8/JaGFvuJiTYlVX4qLzafFyXYhI2GUQoi5o/kyIBKenRehAJgGyQGb
KkTwFbGfXPb/Soajle537IEYFcYLoz+RrX6zfIOZYaFZmhSVsx5uPtPrOxbGuM89NQWplv2OFYvGJ5WNhAdes3rwML53lYJIdBdJ
ehAtlzcitB8IJEO2KvuORgoRC7GuJDLaDRAcmmHAQ37c75VWpFypfoyRPDZvkKO3bueV8U3DdLjutuOUpB2jsb/zMnX3uTgrX255
zNxdWB5SUDO4n/9lfMChFzQpcNEhJ7eeMdFxeFddsy+VEFGQ6AU= root@prasad
```

Ssh-copy-id

```
root@prasad:~# ssh-copy-id root@192.168.234.60
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '192.168.234.60 (192.168.234.60)' can't be established.
ECDSA key fingerprint is SHA256:etTOBRMAV+sQ9O+JJ/XxJsV6Ndd8/2SkEjeK7gjz+WM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alr
eady installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to inst
all the new keys
Password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@192.168.234.60'"
and check to make sure that only the key(s) you wanted were added.
```

TRY TO GET SSH ACCESS PASSWORDLESS

```
root@prasad:~# ssh root@192.168.234.60
Last login: Sun Sep  1 11:08:09 2024
Have a lot of fun...
localhost:~ #
```

Verify the copy ssh key in file ~/.ssh/authorized_keys

```
localhost:~ # cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCvwTwHzIvCsszQ0rNo2cPNNl/AoW55IivV0I29uCS/A17m9KwC0YJ9vtkjI/65
VmT17QmwTthOpGxpsDS/hkARthb/fkjT2FfxcEAvhhqz7sl9/3yxOqLXBmNxGsPCL4/+eJNe70I+kwoNfVB/pPpRRYRkrDrTdFCm
xKJuJFsNrY+LbE4J0YKBf6HH9XP0H8ymkz8BI521dXBkd+8/JaGFvuJiTYlVX4qLzafFyXYhI2GUQoi5o/kyIBKenRehAJgGyQGb
KkTwFbGfXPb/Soajle537IEYFcYLoz+RrX6zfI0ZYaFZmhSVsx5uPtPrOxbGuM89NQWplv2OFYvGJ5WNhAdes3rwML53lYJIdBdJ
ehAtlzcitB8IJE0ZKvuORgoRC7GuJDLaDRAcmmHAQ37c75VWpFypfoyRPDZvkK03bueV8U3DdLjutuOUpB2jsb/zMnX3uTgrX255
zNxdWB5SUDO4n/9lfMChFzQpcNEhJ7eeMdFxeFddsy+VEFGQ6AU= root@prasad
localhost:~ #
```

CONCLUSION

- Summary of Key-Based Authentication
  - Secure, efficient, and password-free authentication
  - Essential for protecting sensitive data
- Best Practices
  - Generate strong keys
  - Manage private keys carefully
  - Configure SSH appropriately


THANK
YOU!