Who am I?

- Bharath Sadashivaiah
- Currently MSc Cyber Security Engineering, WMG.
- Interested in cryptography and privacy-enhancing technologies (Zero-knowledge).
- Previous work experience encompasses software development, systems architecture, and focusing on high-availability solutions.
- Security Engineering (PaaS) and ShadowIT.
- Roles: asg => ops + dev = (devops += tester) U => devSecops ∑ SRE

SSH Jumphost

(DMZ solution to access remote machines)

Index

- Who am I ?
- What is SSH ?
- How to login using SSH?
- Certificate and Tunneling.
- System Diagram.
- Sequence of login.
- SSH Details.
- Demo.
- Tech Stack and Future scope.
- Q n A.
- Reference.

What is SSH?

- The Secure Shell Protocol (SSH) is a cryptographic network protocol for operating network services securely over an unsecured network.[1]
- Its most notable applications are remote login and command-line execution.
- Works on TCP/IP (port 22), using Public key cryptography.
- Supports SSH tunneling, or 'port forwarding' (this will be leveraged to implement the system).
- Popular tool openssh, (openssl = openssh + cryptographic library). Also client tools Putty, SecureCRT, wolfSSH, Dropbear, etc.

How to login using SSH?

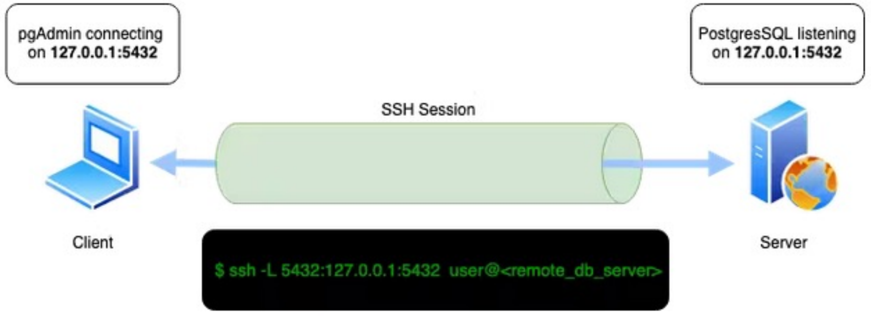
- Password based logins
  + ssh username@host/ip/dns:/landing_path , password on promte.
- Public Key based logins
  + ssh -i key.file username@host/ip/dns:/landing_path

This can be leaked/cracked.

Other methods: Central Authentication Service(CAS) like LDAP, Kerberos, AD(Active Directory). Example product Vintela Authentication Services (VAS)[2].

Certificate and Tunneling/Port Fwd

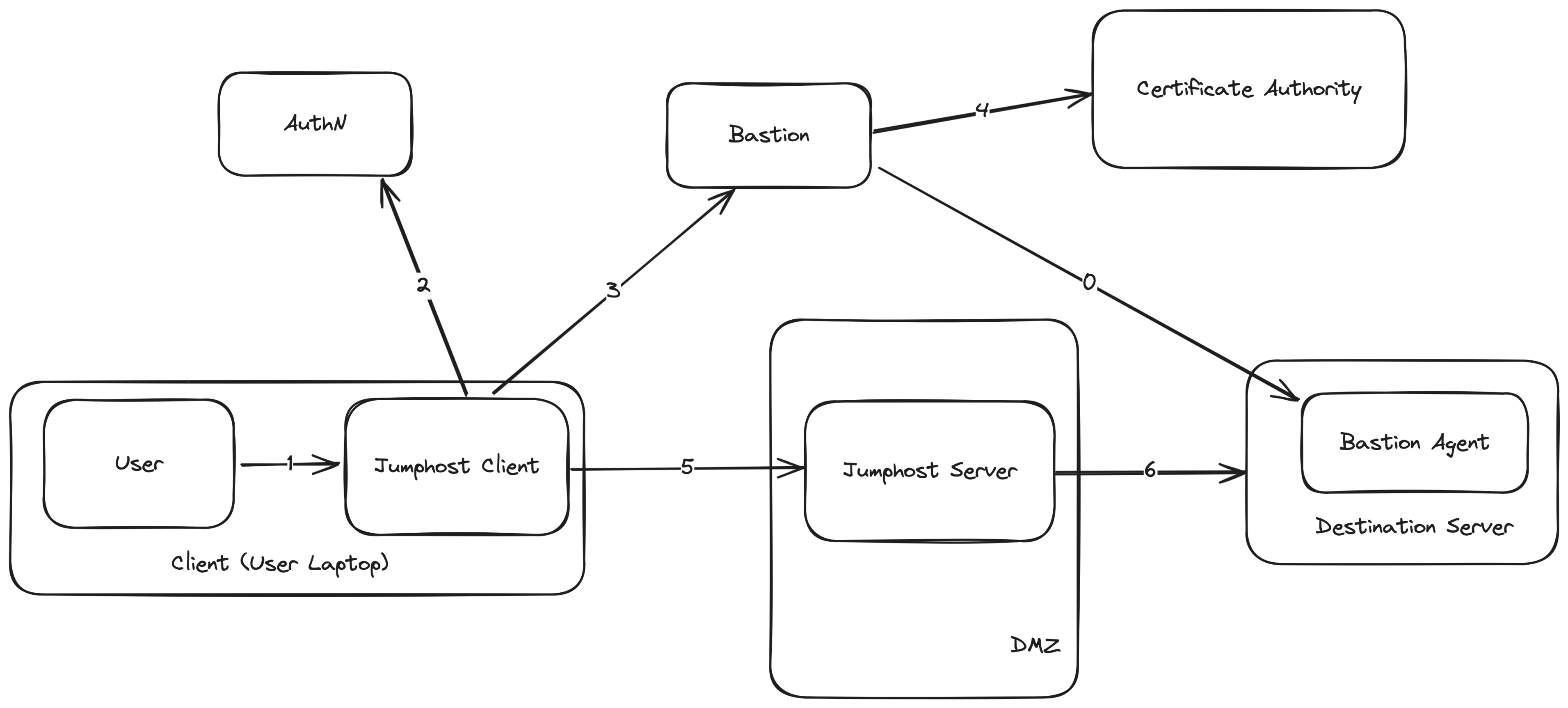
- Scalable and Secure – Can add more users with tested public key certificate.
- Role based access(RBAC) – Fine grain access.
- Ephemeral and Flexible – Short lived and easy to rotate if compromised.
- Tunneling and Port Fwd- Secure access to remote services (e.g., databases, web servers), including dynamic port forwarding (SOCKS proxy).


pgAdmin connecting on 127.0.0.1:5432
PostgresSQL listening on 127.0.0.1:5432
SSH Session
Client
Server

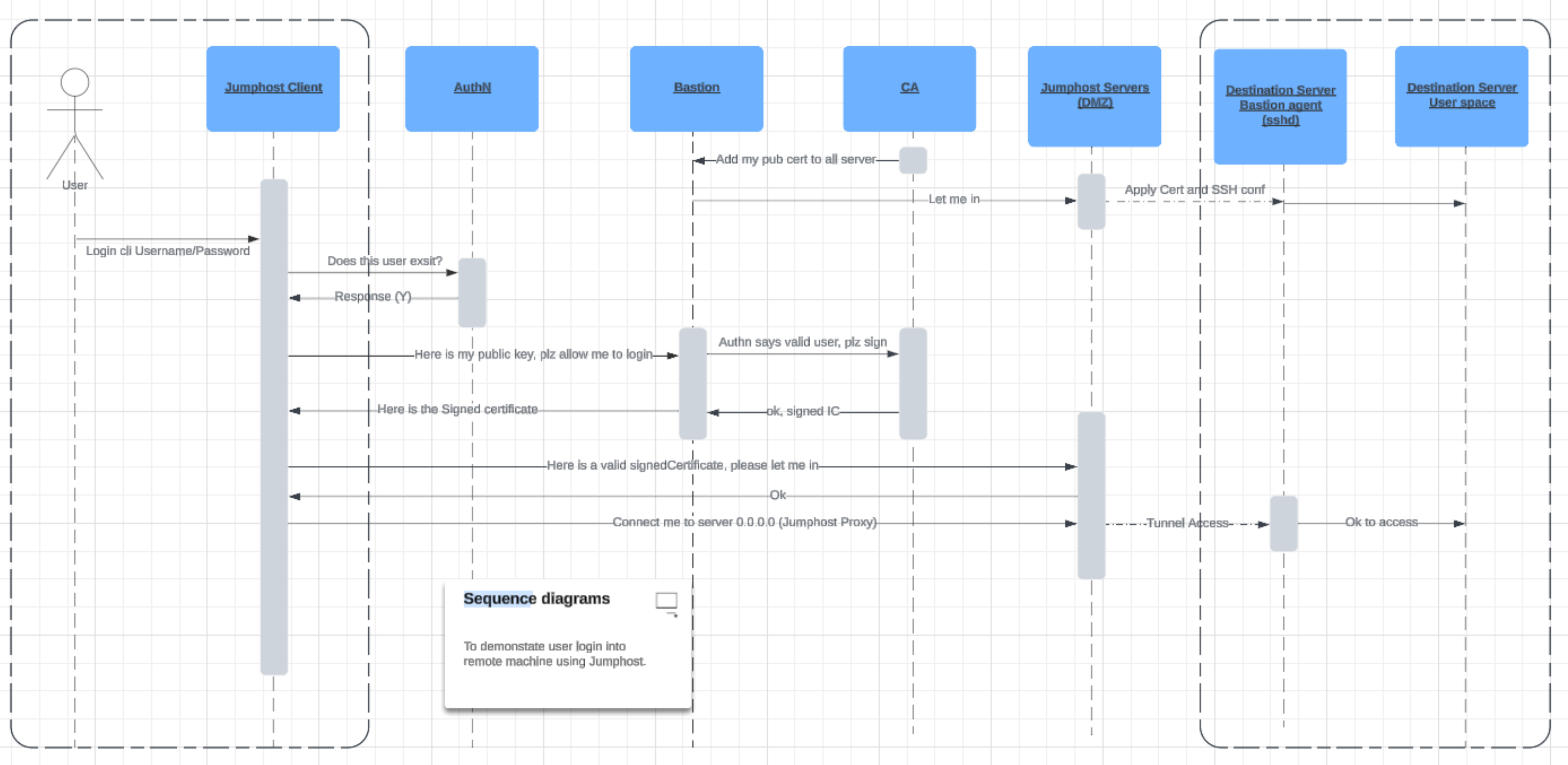
```
$ ssh -L 5432:127.0.0.1:5432  user@<remote_db_server>
```

Fig: SSH local port forwarding.

System Diagram



Sequence of login



User

Jumphost Client

AuthN

Bastion

CA

Jumphost Servers (DMZ)

Destination Server Bastion agent (sshd)

Destination Server User space

Add my pub cert to all server

Let me in

Apply Cert and SSH conf

Login cli Username/Password

Does this user exsit?

Response (Y)

Here is my public key, plz allow me to login

Authn says valid user, plz sign

ok, signed IC

Here is the Signed certificate

Here is a valid signedCertificate, please let me in

Ok

Connect me to server 0.0.0.0 (Jumphost Proxy)

Tunnel Access

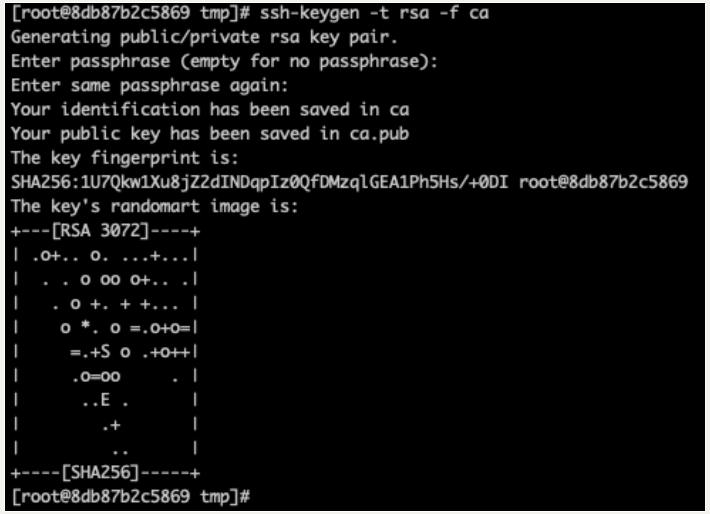
Ok to access

Sequence diagrams

To demonstate user login into remote machine using Jumphost.

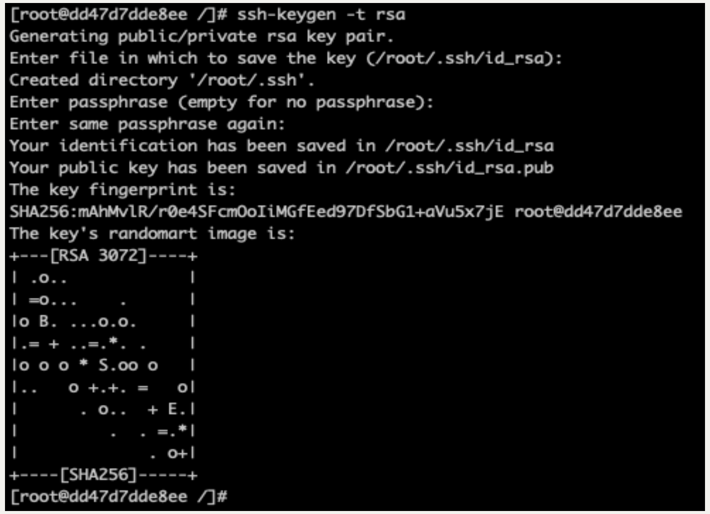
SSH Details and Screenshots

CA Key pair



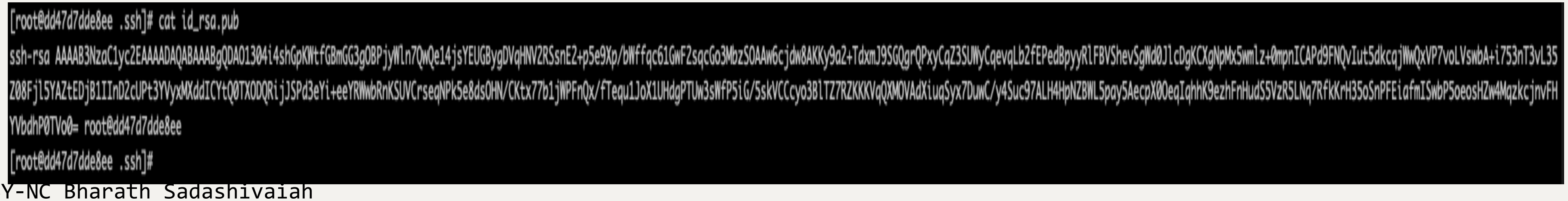
```
[root@8db87b2c5869 tmp]# ssh-keygen -t rsa -f ca
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ca
Your public key has been saved in ca.pub
The key fingerprint is:
SHA256:1U7Qkw1Xu8jZ2dINDqpIz0QfDMzqlGEA1Ph5Hs/+0DI root@8db87b2c5869
The key's randomart image is:
+---[RSA 3072]----+
| .o+.. o. ...+...|
|  . . o oo o+.. .|
|   . o +. + +... |
|    o *. o =.o+o=|
|     =.+S o .+o++|
|     .o=oo     . |
|      ..E .      |
|        .+       |
|         ..      |
+----[SHA256]-----+
[root@8db87b2c5869 tmp]#
```

User key pair

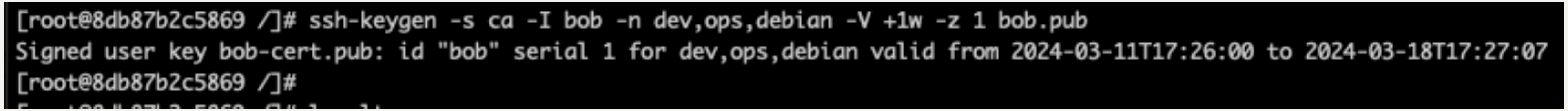


```
[root@dd47d7dde8ee /]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:mAhMvlR/r0e4SFcmOoIiMGfEed97DfSbG1+aVu5x7jE root@dd47d7dde8ee
The key's randomart image is:
+---[RSA 3072]----+
| .o..            |
| =o...    .      |
|o B. ...o.o.     |
|.= + ..=.*. .    |
|o o o * S.oo o   |
|..   o +.+. =   o|
|      . o..  + E.|
|         .  . =.*|
|             . o+|
+----[SHA256]-----+
[root@dd47d7dde8ee /]#
```

User public key



```
[root@dd47d7dde8ee .ssh]# cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDAO1304i4shGpKWtfGBmGG3gOBPjyWln7QwQe14jsYEUGBygDVqHNV2RSsnE2+p5e9Xp/bWffqc61GwF2sqcGo3MbzSOAAw6cjdw8AKKy9a2+TdxmJ9SGQgrQPxyCqZ3SUWyCqevqLb2fEPedBpyyRlFBVShevSgWd0JlcDgKCXgNpMx5wmlz+0mpnICAPd9FNQvIut5dkcqjWwQxVP7voLVswbA+i753nT3vL35
Z08Fjl5YAZtEDjB1IInD2cUPt3YVyxMXddICYtQ0TXODQRijJSPd3eYi+eeYRWwbRnKSUVCrseqNPk5e8dsOHN/CKtx77b1jWPFnQx/fTequ1JoX1UHdgPTUw3sWfP5iG/5skVCCcyo3BlTZ7RZKKKVqQXMOVAdXiuqSyx7DuwC/y4Suc97ALH4HpNZBWL5pay5AecpX0OeqIqhhK9ezhFnHudS5VzR5LNq7RfkKrH35oSnPFEiafmISwbP5oeosHZw4MqzkcjnvFH
YVbdhP0TVo0= root@dd47d7dde8ee
[root@dd47d7dde8ee .ssh]#
```



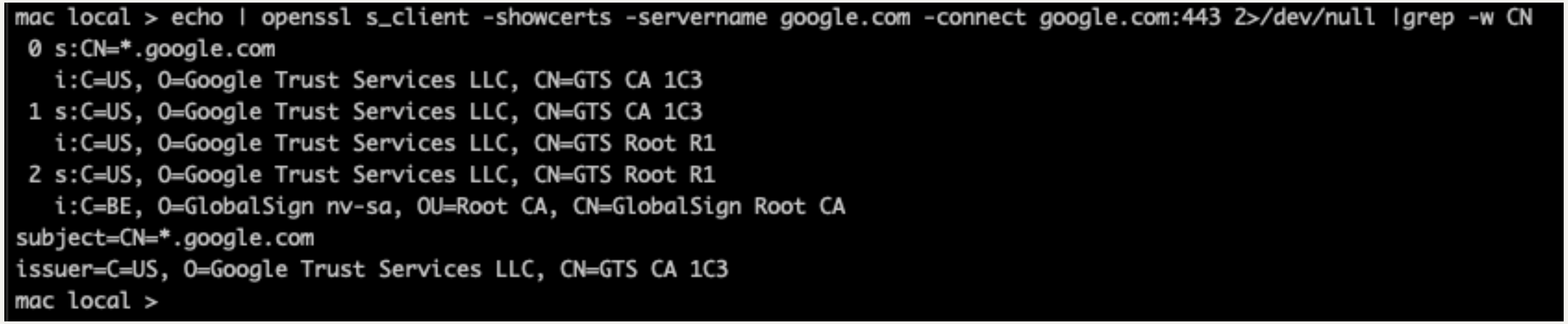
```
[root@8db87b2c5869 /]# ssh-keygen -s ca -I bob -n dev,ops,debian -V +1w -z 1 bob.pub
Signed user key bob-cert.pub: id "bob" serial 1 for dev,ops,debian valid from 2024-03-11T17:26:00 to 2024-03-18T17:27:07
[root@8db87b2c5869 /]#
```

What are we signing

- “-s ca” signing using the server certificate.
- “-I bob” identity certificate for user.
- “-n dev,ops,debian” principles limitation, users/hosts.
- “-V +1w” validity period

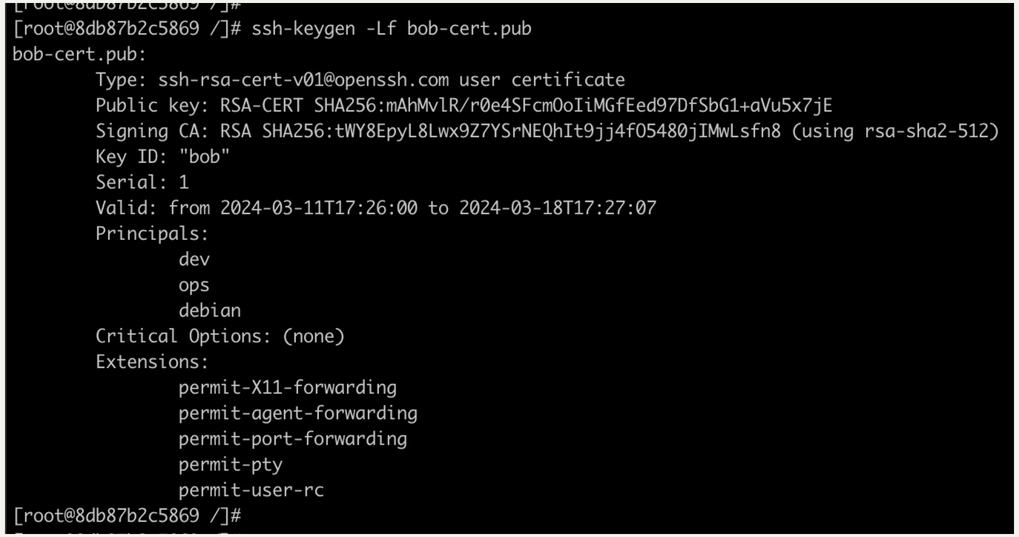
Further details in https://man.openbsd.org/ssh-keygen

Certificate Chain



```
mac local > echo | openssl s_client -showcerts -servername google.com -connect google.com:443 2>/dev/null |grep -w CN
 0 s:CN=*.google.com
   i:C=US, O=Google Trust Services LLC, CN=GTS CA 1C3
 1 s:C=US, O=Google Trust Services LLC, CN=GTS CA 1C3
   i:C=US, O=Google Trust Services LLC, CN=GTS Root R1
 2 s:C=US, O=Google Trust Services LLC, CN=GTS Root R1
   i:C=BE, O=GlobalSign nv-sa, OU=Root CA, CN=GlobalSign Root CA
subject=CN=*.google.com
issuer=C=US, O=Google Trust Services LLC, CN=GTS CA 1C3
mac local >
```

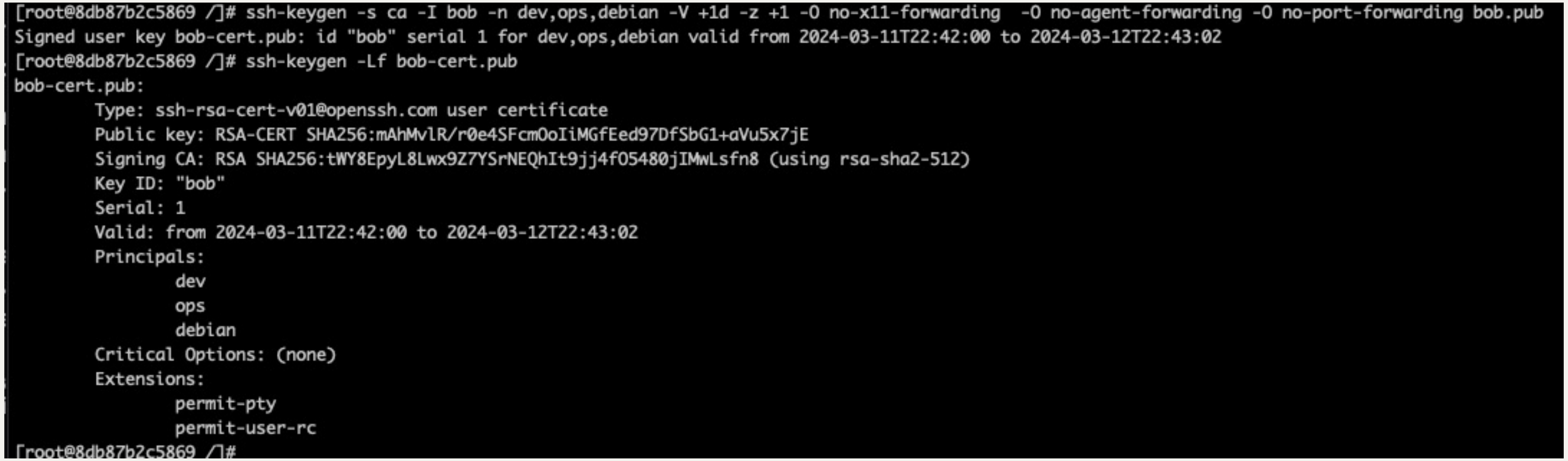
SSH Certificate Details



```
[root@8db87b2c5869 /]#
[root@8db87b2c5869 /]# ssh-keygen -Lf bob-cert.pub
bob-cert.pub:
        Type: ssh-rsa-cert-v01@openssh.com user certificate
        Public key: RSA-CERT SHA256:mAhMvlR/r0e4SFcmOoIiMGfEed97DfSbG1+aVu5x7jE
        Signing CA: RSA SHA256:tWY8EpyL8Lwx9Z7YSrNEQhIt9jj4f05480jIMwLsfn8 (using rsa-sha2-512)
        Key ID: "bob"
        Serial: 1
        Valid: from 2024-03-11T17:26:00 to 2024-03-18T17:27:07
        Principals:
                dev
                ops
                debian
        Critical Options: (none)
        Extensions:
                permit-X11-forwarding
                permit-agent-forwarding
                permit-port-forwarding
                permit-pty
                permit-user-rc
[root@8db87b2c5869 /]#
```

Lots of access

Limited access



```
[root@8db87b2c5869 /]# ssh-keygen -s ca -I bob -n dev,ops,debian -V +1d -z +1 -O no-x11-forwarding  -O no-agent-forwarding -O no-port-forwarding bob.pub
Signed user key bob-cert.pub: id "bob" serial 1 for dev,ops,debian valid from 2024-03-11T22:42:00 to 2024-03-12T22:43:02
[root@8db87b2c5869 /]# ssh-keygen -Lf bob-cert.pub
bob-cert.pub:
        Type: ssh-rsa-cert-v01@openssh.com user certificate
        Public key: RSA-CERT SHA256:mAhMvlR/r0e4SFcmOoIiMGfEed97DfSbG1+aVu5x7jE
        Signing CA: RSA SHA256:tWY8EpyL8Lwx9Z7YSrNEQhIt9jj4f05480jIMwLsfn8 (using rsa-sha2-512)
        Key ID: "bob"
        Serial: 1
        Valid: from 2024-03-11T22:42:00 to 2024-03-12T22:43:02
        Principals:
                dev
                ops
                debian
        Critical Options: (none)
        Extensions:
                permit-pty
                permit-user-rc
[root@8db87b2c5869 /]#
```

Demo

Two Docker containers:

- ssh client (dev laptop)
- ssh server (CA + destination server)

Tech Stack

- JumpHost Client: API any language.
- AuthN: OAuth.
- Bastion: Backend application API any language.
- Certificate Authority: Hashicorp Vault.
- Jumphost Server: openssh.
- End Server: sshd service.

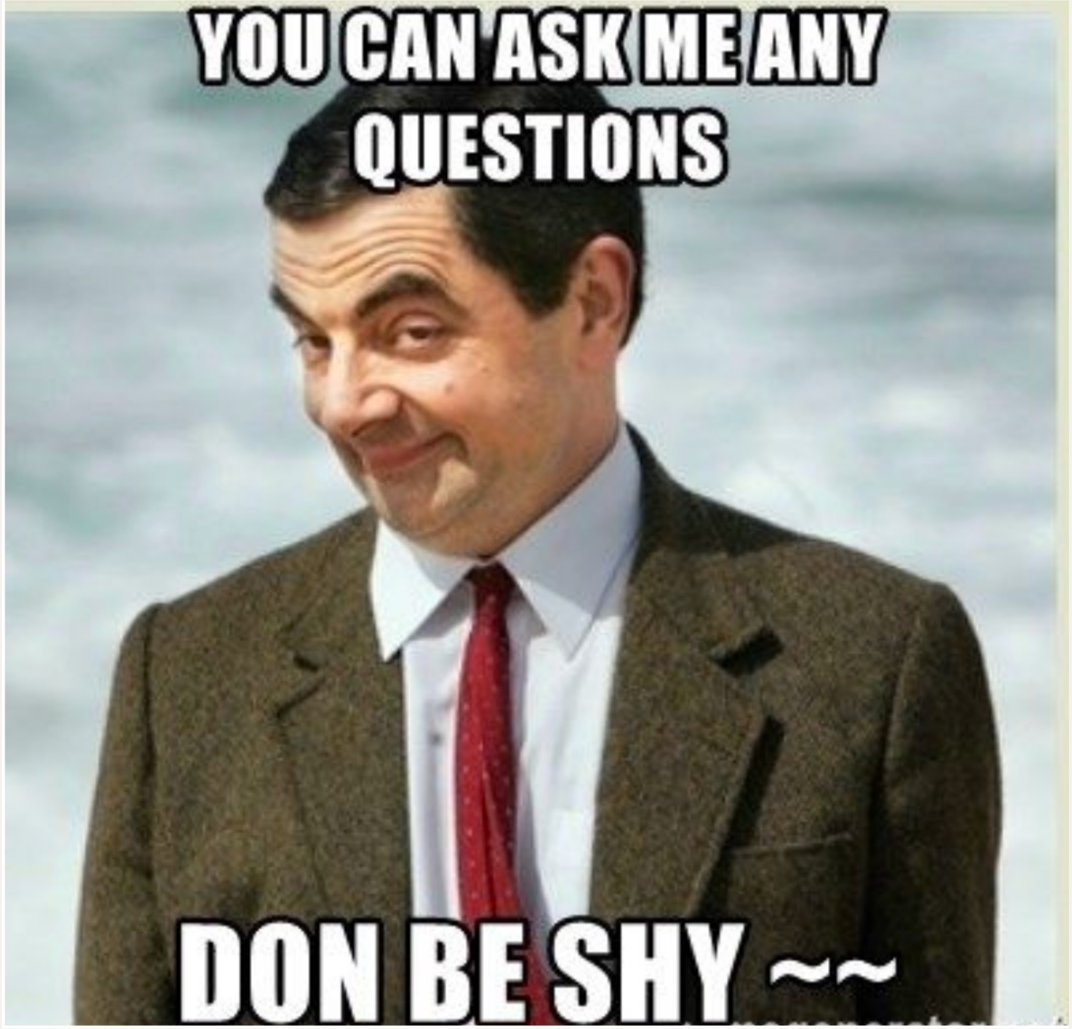
Other Interesting Tools:

- https://goteleport.com/
- https://www.okta.com/uk/

Future Improvements

- Remove Username/password, passkeys.
- SSH supports FIDO implementation.
- Auditable and transparent connections (Teleport).
- https://goteleport.com/blog/introducing-teleport-4-point-3-modern-replacement-for-openssh/

THANK YOU FOR LISTENING
ANY QUESTIONS?


YOU CAN ASK ME ANY QUESTIONS
DON BE SHY ~~

Reference

1. https://datatracker.ietf.org/doc/html/rfc4251
2. https://www.oneidentity.com/
3. https://github.com/Netflix/bless
4. https://github.com/uber/pam-ussh