Certificate-based SSH authentication

1 - Abstract

Certificate-based SSH authentication is superior to SSH keys in many ways:

- SSH certificates intrinsically possess a validity period before and after which they are invalid for providing authentication.
- SSH certificates can be embedded with SSH restrictions that limit:
    - Who can use the certificate
    - The list of available SSH features (X11Forwarding, AgentForwarding, etc)
    - Which SSH client machines can use the certificate
    - Commands that can be run via SSH

2- Repository

The following github repository provides the code base to setup a Certification Authority and later sign the certificates.

> https://github.com/jlangenegger/ssh_certificate/

3 - Setup

For the purposes of this explanation, let's consider three systems:

- Certification Authority (CA)
    - System name "ca.netdef.org"
    - Will host our Certification Authority
- Host
    - System name "host.netdef.org"
    - Will function as an SSH server
- Client
    - System name "client.netdef.org"
    - Will function as an SSH client

4 - Certificates

There are two different certificates that are possible:

- client certificate
    - This certificate is stored on the client and is provided to the host during the ssh connection establishment.
    - It is used on the host side to authenticate the clients that try to login.
    - This certificate replaces public key or password based login.
- host certificate
    - This certificate is stored on the host and is provided to the client during the ssh connection establishment.
    - It is used on the client side to authenticate the host that the client tries to login.
    - This certificate replaces the authorized key file entry for a given host.

Here at NetDEF we use the client certificate only.

5 - Configuration

There are separate pages the guide you through the installation process for the Certificate Authority, the client and the host:

- Certificate Authority
- `Client Setup`
- `Host Setup`