Design

Overview

Network Fandango (NF) is a combination of traditional network management suite and Infrastructure-as-Code system. The main components are:

- nfsvc - the main service.
- nf - command line tool which connects to the main service.
- nfweb - web front end.
- nfagent - a hypothetical agent which can be installed on machines. This part may be unnecessary.

In addition, there will be a scripting language which will allow the definition of playbook actions.

Network Discovery

Network discovery occurs in a number of different ways. For administrators with control over their own DNS infrastructure, we recommend using either of the following records. For these examples, we are using example.com as your domain.

```
_nfsvc.example.com   IN  A   192.168.1.1
```

or

```
_nfsvc._tcp.example.com   IN   SRV    0 0 9626 server.example.com
```

In addition to these methods, other methods will exist, such as SSDP but perhaps not until further into the development.

Certificates and their Uses

Network Fandango Service

The following certificates are generated by the Network Fandango Service:

Root CA

- X509 Certificate
- Trust anchor for all service issued TLS certs and JWT tokens. You can optionally use your own organisation's certificate infrastructure for this certificate.
- Issued By: `ICertificateService.CreateRootCertificateAsync`
- Obtain With: `ICertificateService.TryLoadRootCa` or `ICertificateService.TryGetRootPublicKey`
- Storage Keys: `ca.x509.root.pem`, `ca.x509.root.key`

API TLS Certificate

- X509 Certificate
- HTTPS endpoint identity for nfsvc
- Issued By: `ICertificateService.IssueTlsCertificateAsync`
- Obtain With: `ICertificateService.TryLoadTlsCertificateAsync`
- Storage Keys: N/A (uses filesystem)

JWT Certificate

- X509 Certificate
- Sign server-issued JWTs (enrollment tokens, etc.).
- Issued By: `ICertificateService.IssueJwtCertificateAsync`
- Obtain With: `ICertificateService.TryLoadJwtCertificateAsync`
- Storage Keys: N/A (uses filesystem)

SSH User CA (OpenSSH CA for user certs)

- Sign user SSH certificates for ephemeral admin access.
- Issued By: `ICertificateService.IssueSshUserCertificateAsync`

- Obtain With: `ICertificateService.TryGetSshUserPublicKey`
- Storage Keys: `ssh_ca_key`, `ssh_ca_key_pub`

SSH Host CA (OpenSSH CA for host certs)

- Sign host SSH certificates for devices so admins can verify host identity without managing known_hosts entries.
- Issued By: `ICertificateService.IssueSshHostCertificateAsync`
- Obtain With: N/A
- Storage Keys: `ssh_host_ca_key`, `ssh_host_ca_key_pub`

Enrolled Network Fandango Hosts

Enrolled hosts download the Root CA public key and SSH public keys at the point of enrollment, in order to verify the identity of `nfsvc`, but these certificates are not generated on the enrolled hosts.

mTLS Client Certificate

- X509 Certificate
- Identify the device to nfsvc over mutual TLS (agent calls, jobs, facts).

Authentication

The authentication flow is modeled on that of SSH, specifically using keyfiles. The steps involved in authentication are as follows:

1. Install Network Fandango
2. Set up your Network Discovery
3. The first time the Network Fandango service (`nfsvc`) is started, it will create a root CA certificate, a TLS certificate and a JWT certificate.
4. You must copy the root CA certificate (`ca.pem`) to your workstation. Put it in `~/.nf/certs/[server]`
5. On your workstation, install the Network Fandango client tools
6. On your workstation, run `nf`

When you do this, `nf` will:

- Discover your server
- Verify that the server's TLS certificate is signed by the root CA certificate you copied to your workstation
- Create a public and private key file (in `~/.nf/certs/[server]`)
- Request a nonce from the server
- Sign the nonce using the private key
- Submit the signed nonce, public key and your username to the server

Because this is the first login, the key will be accepted without question, and the initial user account created. Your public key will be stored for future use.

The server will respond with a JWT token, signed with the JWT certificate. `nf` will verify the JWT against the JWT certificate by requesting it from the server's JWKS endpoint.

`nf` will cache the JWT, which are valid for 8 hours by default.

On subsequent requests to the server, `nf` will detect that the JWT exists and attempt to verify it. If it fails, it will repeat the above process. The difference is that the provided public key must match the previously provided one because this is not the first login.

Subsequent users must be made manually by the initial user. You have two options for doing this:

1. The first time the user logs in, their key is accepted without quesion.
2. You set a one time password on the new user, and the first time the user connects, they must provide this or their key will not be accepted. The password is not required for future use of the `nf` command line tool.

The authentication flow for the web app has not yet been designed.

The nf command

01 The Basics

Common Parameters

All Commands

These parameters work with all commands.

```
--json
```

Outputs the result in JSON.

Item Lists

For commands which return lists of items (users, computers, etc), you can use the following parameters:

```
-p <int>
```

Retrieve the specified page number of results. The page number defaults to 1.

```
-s <int>
```

Set the page size. The number of pages will therefore be `number of items` / `page size`. The default page size is 15.

```
--search <terms>
```

Filter the list of items returned. The exact filtering methodology can vary by command, but typically all properties are searched.

```
--format <format string>
```

Format the output in a specific way. This exists to make it easier to pipe the output to another command. Using this will output only the resulting items, and none of the additional information

you might otherwise see. The way to use this is:

```
--format "%{i} %{username} %{otherProperty}"
```

You can use `%{i}` to access the current row number. This will reset to 1 for each page.

Otherwise, you can use any of the properties defined on the objects being returned. In this case, `username` and `otherProperty`. Property names are not case sensitive.

02 Authentication

Logging in as the first user on a new installation

To log in under these circumstances, all you have to do is run `nf auth`. It will create a key file which the server will automatically accept as there are no configured users.

Logging in as a previously created user

Simply run `nf` with whatever parameters you need to achieve your goals. You will be logged in automatically.

If you are using a different user name than you were when you set up your account, you can use `nf set --user <username>` then use `nf` as normal.

Logging in as a new user on an existing system

This depends on how your account was set up.

03 User Management

Create a user

```
nf user add <username>
```

Creates a user with a randomised initial login token. The token will be provided in the command output, and you can pass this on to the user for use on their first login.

```
nf user add <username> --initial-token <token>
```

Creates a user with a manually set initial login token.

```
nf user add <username> --no-token
```

Creates a user with no initial login token. This means that the server will accept the first login to the user account with minimal checks. **Use this option with caution.**

List Users

```
nf user list [-p <page>] [-s <pageSize>] [--search <search terms>]
```

```
nf user ls [-p <page>] [-s <pageSize>] [--search <search terms>]
```

Lists the configured users along with their creation and last login times.

You can use the standard page, page size and search parameters.

Set a Password

> This functionality is not guarateed to exist prior to version 1.0, and is subject to change.

```
nf user password <user> <password>
nf user password <user> --random
```

```
nf user pw <user> <password>
nf user pw <user> --random
```

Command line users do not need passwords, instead relying on keyfiles. You can, however, set a password for a user, which will allow them to log in to the web interface.

This has no bearing on command line users.

Delete a user

```
nf user delete <username>
```

Deletes the specified user.

Database Schema

Computers