SSH Certificates and Certificate Management

And a shameless plug for a personal project

Mike Lovell – mike@dev-zero.net
OpenWest 2017

http://openwest.dev-zero.net/openwest-2017-ssh-certificates.pdf

Review of SSH Keys

- Easier, more secure authentication of User to a Host
- Validates a Host to a User

```
ssh-keygen -f /path/to/private-key

ssh-copy host

ssh -i /path/to/private-key host
```

Review of ssh-agent

- Provides some simplicity around keys that are encrypted on disk
- ssh-agent holds the private key in memory, listens on a Unix Socket, responds to requests to add/remove keys and sign data

```
eval `ssh-agent`

ssh-add /path/to/private-key

ssh -A host
```

Pitfalls

- Individual keys for Users have to be distributed to Hosts
- Host key validation is essentially non-existent
  - Possible if something gathers them (ssh-keyscan) or stored in DNS (SSHFP records)
  - Neither are common
- Compromise of a key requires removal of the key from all Hosts
  - No expiration
- Keys don’t include constraints themselves
  - They’re possible though simple editing of a text file

SSH Key Management

```
ssh_authorized_key { ‘mike ssh key’:
    User => ‘mike’,
    Ensure => present,
    Type => ‘ssh-rsa’,
    Key => ‘AAAAB3...s0oQ==’
}
```

SSH Key Management

```
- name: ‘mike ssh key’
  authorized_key:
    User: mike
    State: present
    Key: ssh-rsa AAAAB3...s0oQ==
```

SSH Key Management

Wikimedia Keyholder
https://blog.wikimedia.org/2017/03/22/keyholder/

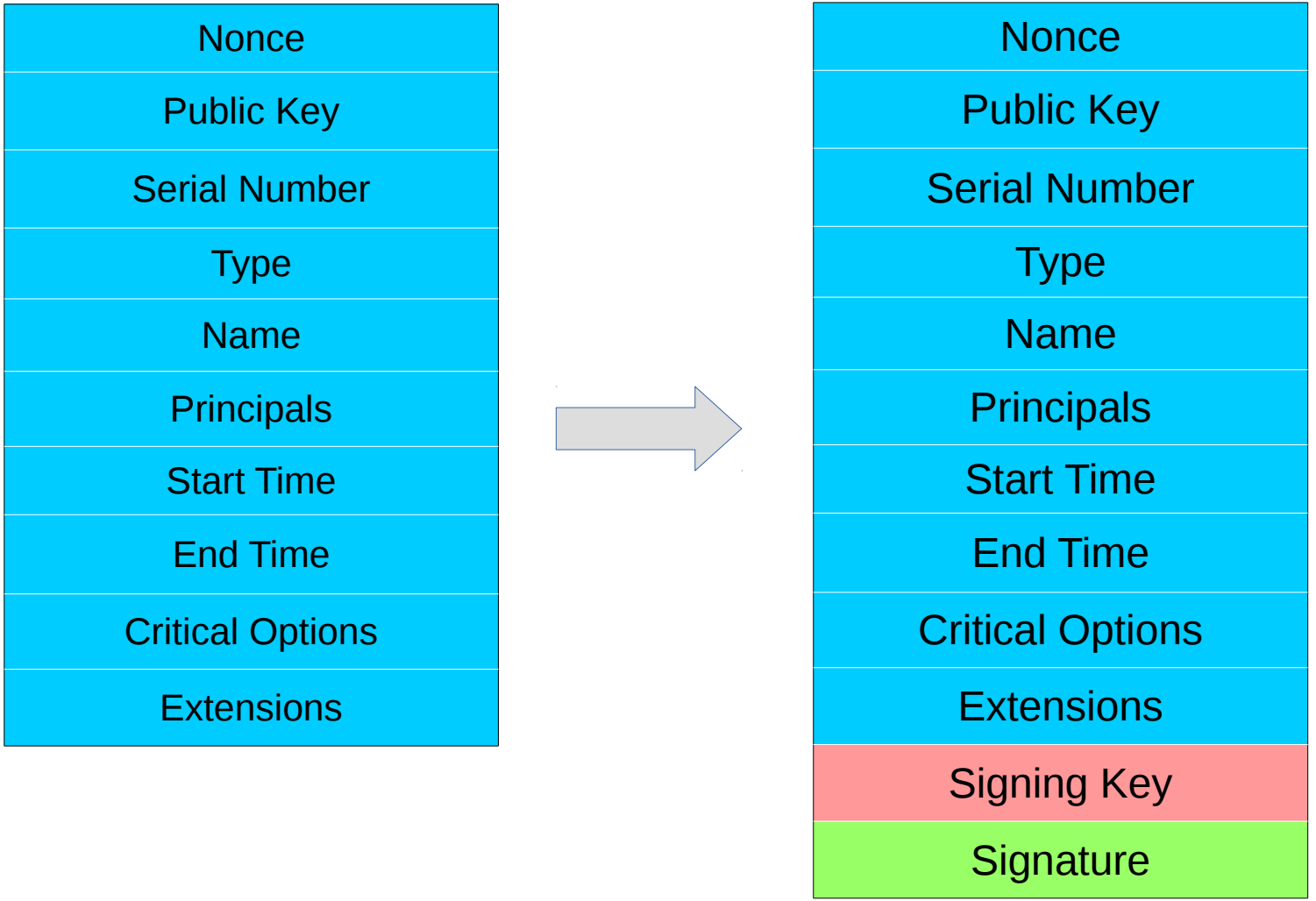
SSHecret
https://github.com/thcipriani/sshecret

Kryptonite:
https://krypt.co/

OpenSSH Certificates

Certificates – Basic Concept

- Gather several bits of information
  - Public Key of a User or Host
  - The User's or Host's name (a.k.a the 'principal')
  - Start and End time
  - Allowed options
  - Serial number
- Sign those bits with a trusted private key (a.k.a the CA key)
- Distribute the CA key to Users and Hosts
- Users and Hosts can validate each other using the CA key and the signed bits (a.k.a the certificate)


Nonce
Public Key
Serial Number
Type
Name
Principals
Start Time
End Time
Critical Options
Extensions
Nonce
Public Key
Serial Number
Type
Name
Principals
Start Time
End Time
Critical Options
Extensions
Signing Key
Signature

Certificates – Basic Config

- Create a CA key

```
ssh-keygen -f /path/to/ca-key
```

- Add to known_hosts or authorized_keys
  - Use @cert-authority at beginning of known_hosts or cert-authority for authorized_keys
- Sign a Host's public key

```
ssh-keygen -s /path/to/ca-key -h -n
host.domain /path/to/host_key.pub
```

- Sign a User's public key

```
ssh-keygen -s /path/to/ca-key -n username
/path/to/id_rsa.pub
```

Demo

Certificates – Pitfalls

- CA public key has to be distributed everywhere
- Creation of a certificate requires access to the CA private key
- ssh-keygen doesn’t have any policies for certificates
  - Doesn’t check ‘principals’
  - No enforcing of date validity
  - User can specify any options

Introducing Janus

https://www.github.com/mikelovell/janus

Janus - Intro

- Python code that only has a few dependencies
  - paramiko, ecdsa, and, cryptography
  - falcon, passlib, eventlet
  - requests, prettytable
- Manages Certificate Authorities (Yes, more than one)
- Can apply policy filters on requests to limit what a user can request
- Requests made through HTTP API
  - Local requests also available but there are risks

Janus – Brief Tour

- janus-cli
  - CLI utility for directly managing Authorities
  - Can provide information about configured Cas
  - Has a `serve` function to run the HTTP API
- janus
  - CLI utility for accessing the HTTP API
  - Not required but makes things simpler

Janus – Brief Tour

- Configuration file
  - INI style file read by ConfigParser
  - Should configure at least one Authority
  - Each Authority needs a Datastore, Key Backend, and a List of Policy Filters

Janus Demo

Janus – Work to do

- Documentation
- Only the CA Listing and Certificate Requests parts of the API are implemented
  - No listing of certs
  - Delayed signing not yet implemented
- Host key signing not implemented
- Key Revocation not even started. Need a Python implementation of the OpenSSH KRL.
- Need more work on filters
- Database Datastore