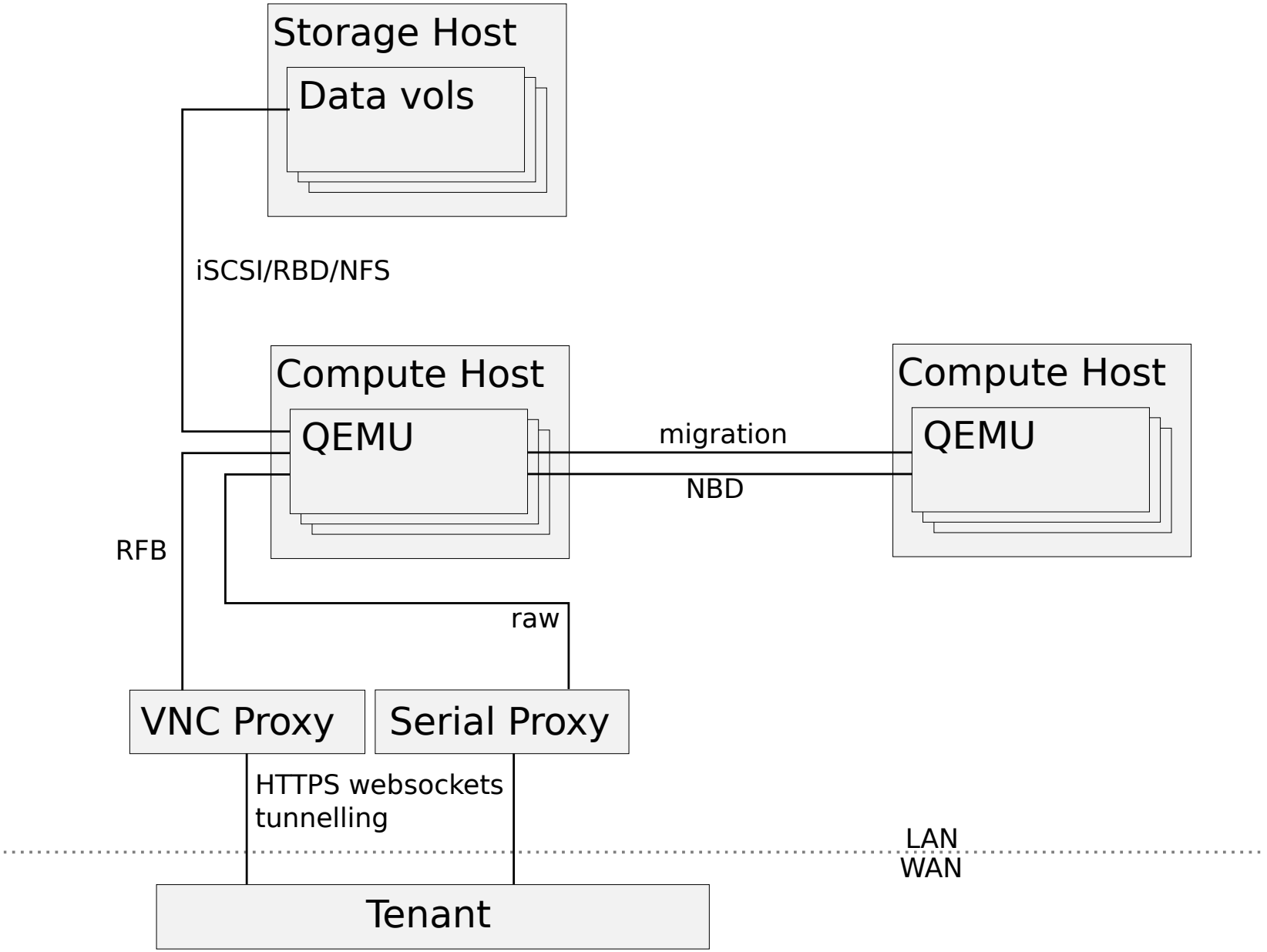
Securely integrating with QEMU

Guidance for Open Source virtualization developers

Daniel P. Berrangé
@
Red Hat Cloud Engineering

Scope of talk

- Host management security eg libvirt, openstack
- Not QEMU/KVM guest ABI security
- Securing the management LAN
- Secrets, encryption and authorization


Storage Host
Data vols
iSCSI/RBD/NFS
Compute Host
QEMU
migration
NBD
Compute Host
QEMU
RFB
raw
VNC Proxy
Serial Proxy
HTTPS websockets
tunnelling
LAN
WAN
Tenant

Secrets: old

- Adhoc method for each scenario
  - Clear text inline in argv
  - Out of band via monitor
  - Prompt on console
- Complex for mgmt app
- Complex for QEMU maintainers

Secrets: new

- “secret” object type (v2.6.0)
  - raw or base64
  - plain text or encrypted
  - inline or file
- RBD, iSCSI, CURL, LUKS, x509 cert
  - One global master secret via file
  - Per object secrets inline & encrypted

Secrets: creation

- Password inline as clear text (insecure)

```
-object secret,id=sec0,data=letmein
```

- Password from a clear text file (secure)

```
-object secret,id=sec0,file=passwd.txt
```

- Password from a base64 clear text file (secure)

```
-object secret,id=sec0,file=passwd.b64,format=base64
```

- Password inline as aes256 cipher text (secure)

```
-object secret,id=sec0,file=master.aes \
-object secret,id=sec1,data=CIPHERTEXT,\
        keyid=sec0,iv=NNNNNNNNNNNNNNN,
```

Secrets: usage

- Disks with “password-secret” property

```
-drive driver=rbd,\
    filename=rbd:pool/image:id=myname:\
    auth_supported=cephx,password-secret=sec0
```

- TLS cert with “passwordid” property

```
-object tls-creds-x509,dir=/etc/tls/qemu,\
    id=tls0,endpoint=server,paswordid=sec0
```

TLS: credentials

- “tls-creds-anon” object type (v2.5.0)
  - Anonymous DH-params
  - Insecure, only for back compat with legacy VNC VeNCrypt
- “tls-creds-x509” object type (v2.5.0)
  - directory to x509 PEM files
  - optionally require client certs
  - secret key passwords (v2.6.0)
  - cipher priority (v2.7.0)

TLS: credentials

- TLS setup mistakes very hard to diagnose
  - Horrible “handshake failed” error at runtime
- Goal: detect & report mistakes at startup
- Configuration sanity checking
  - Basic constrainbts
  - Key purpose / usage
  - Valid/expiry times
  - Signing chain

TLS: Usage

- All network services
  - VNC (v2.5.0)
  - Character devices (v2.6.0)
  - NBD server/client (v2.6.0)
  - Migration (v2.7.0)
- Credential setup
  - One set of creds per host, or...
  - One set of creds per host, per service, or...
  - One set of creds per host, per network, or..

Disk encryption: intro

- Protect data at rest and data in flight
- LUKS / TrueCrypt inside guest
  - How to provide decryption key at boot
  - Shared key if VM image is cloned
  - Provider can't guarantee security

Disk encryption: old

- qcow2: terminally flawed cryptographic design
  - Hardcoded cipher mode, AES-CBC with plain64 IV
  - Allows watermarking attack
  - Directly encrypted with provided ascii “key”
    - No password change without re-encryption
    - No secure delete without shredding entire volume
- Preventing its ongoing use
  - Deprecated v2.3.0, broken v2.4.0, blocked v2.7.0
  - Use qemu-img / qemu-nbd to liberate data

Disk encryption: new

- LUKS / dm-crypt / cryptsetup
  - Widely adopted, proven & reviewed design, extensible crypto
  - Layer over block device
  - Layer over loopback or qemu-nbd backed devices
  - Requires admin privileges
- QEMU LUKS driver (v2.6.0)
  - Interoperable with dm-crypt/cryptsetup
  - Use with any QEMU block driver
  - Use unprivileged & on any platform
  - QEMU I/O test suite + dm-crypt interop tests

Disk encryption: future

- Key slot management
- Secure deletion
- Integrate into QCow2
- More tunables (e.g. pbkdf2 iters / time)
- Detached header volume
- Performance benchmarking / optimization

Disk encryption: creation

- Provide password via sects

```
--object secret,id=sec0,file=passphrase.b64
```

- Plain file, default parameters (aes-256 + XTS + plain64)

```
qemu-img [SECRET-OPT] create \
    -o key-secret=sec0 \
    -f luks demo.luks 10G
```

- Plain file, custom parameters (aes-256 + CBC + ESSIV)

```
qemu-img [SECRET-OPT] create \
    -o key-secret=sec0,cipher-mode=cbc,ivgen=essiv \
    -f luks demo.luks 10G
```

Disk encryption: creation

- RBD volume

```
qemu-img [SECRET-OPT] create \
    -o key-secret=sec0 -f luks \
    rbd:mypool/demo.img:mon_host=10.73.75.52 10G
```

- QCow2 image (v2.8.0?)

```
qemu-img [SECRET-OPT] create \
    -o key-secret=sec0,encryption=on -f qcow2
    /var/lib/libvirt/images/demo.qcow2 10G
```

Disk encryption: usage

- Plain LUKSv1 file

```
$QEMU -object secret,id=sec0,file=key.bin \
    -drive if=none,driver=luks,key-secret=sec0,\
       id=drive0,file.driver=file \
       file.filename=/var/lib/libvirt/images/demo.luks \
    -device virtio-blk,drive=drive0
```

Disk encryption: usage

- RBD volume

```
$QEMU -object secret,id=sec0,file=key.bin \
    -object secret,id=sec1,file=passwd.bin \
    -drive if=none,driver=luks,key-secret=sec0,id=drive0,\
       file.driver=rbd,file.password-secret=sec1,\
       file.filename=rbd:somehost:9000/somevol \
    -device virtio-blk,drive=drive0
```

Access control: TLS

- x509 based identity verification
- 'verify-peer' requires client x509 cert
- client cert must be signed by CA
- Sub-CA per service to provide access control

Access control: simple

- General framework for plugging authorization systems
- “qauthz-simple” object type (v2.8.0 ?)
  - simple access control lists
  - allow / deny per list entry
  - exact or glob matching
  - global default policy for non-matching
- Integrates with VNC, chardev, migration, NBD (v2.8.0 ?)
  - TLS x509 distinguished name (all)
  - SASL username (VNC only)

Access control: PAM

- “qauthz-pam” object type (v2.8.0 ?)
  - Integrates with PAM for LDAP, SQL DB, & more
- Usage with VNC x509 dname validation

```
-object authz-pam,id=acl0,service=qemu-vnc \
-vnc :1,tls-creds=tls0,tls-acl=acl0
```

  - /etc/pam.d/qemu-acl

```
account requisite pam_listfile.so item=user sense=allow
file=/etc/qemu/vnc.allow
```

  - /etc/qemu/vnc.allow

```
CN=client.foo.acme.com,O=ACME-
Corp,L=London,ST=London,C=GB
```

Your questions ?

Further reading QEMU security blog series

@

https://www.berrange.com/topics/security-2/

Please fill in the KVM Forum Poll:

https://goo.gl/SCCpky