Data Security and Storage Hardening In Rook and Ceph

Federico Lucifredi – IBM | Red Hat

me! me! me!

Things I worked on

Ceph Storage
Ubuntu Server
Landscape
SUSE Studio
SLES
SMT
Ximian Red Carpet
Man (I)

me! me! me!

Things I worked on

Ceph Storage
Incident Response
Computer Security
Formal Methods
Oblivious Computing
Deniable File Systems
Computation on Cellular Automata

me! me! me!

Things I worked on

Red Hat Ceph Storage
Hero of Ceph support
Growing tomatoes
…

CEPH IN BRIEF

Ceph

- The future of storage!
- File, block and object storage
- Highly resilient
- Highly available
- Scale out

...absolutely awesome.

ROOK IN BRIEF

Rook

- Cloud-native storage for k8s
- Ceph-based: hyperscale
- Storage on top of compute: hyper-converge…

  …or optionally external storage

- Automated resource management w/operators
- Automated upgrade and rollback

THREAT MODEL

- Identify threat actors
  - Nation states
  - Organized crime
  - Hacker groups
  - Motivated individuals
  - Privileged insiders
  - Script kiddies
  - …

NETWORK SECURITY ZONES

- Public Zone
  - **not** the public_network in Ceph
- Ceph Client Zone
- Storage Access Zone
  - public_network in Ceph
- Ceph Cluster zone

NETWORK SECURITY ZONES

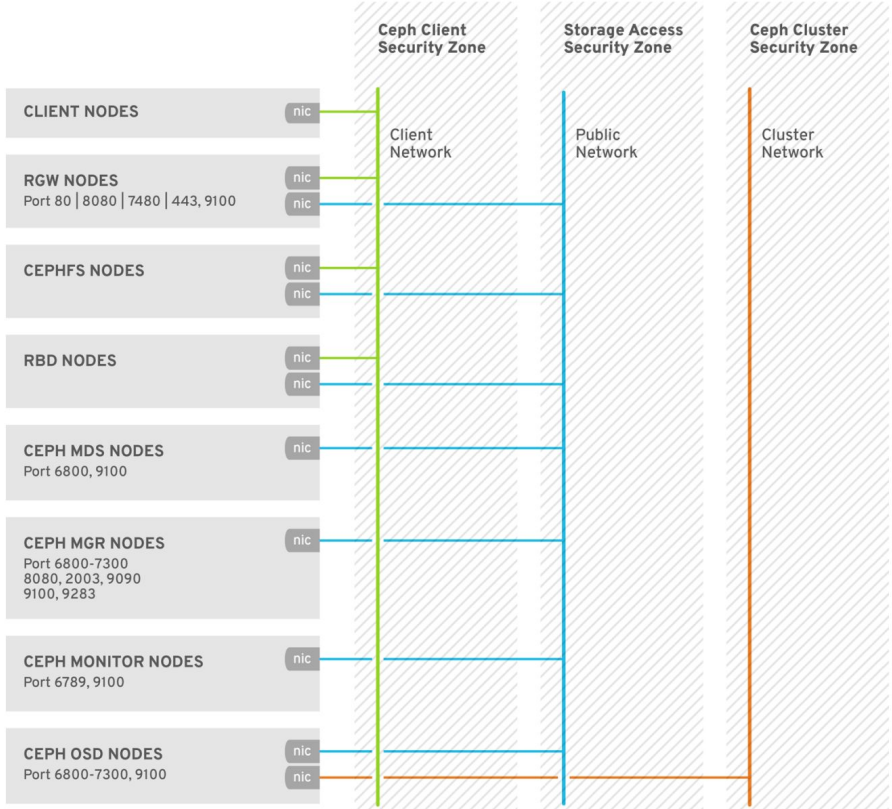
- Public Zone
  - **not** the public_network in Ceph
- Ceph Client Zone
- Storage Access Zone
  - public_network in Ceph
- Ceph Cluster zone

NETWORK SECURITY ZONES

- Public Zone
  - **not** the public_network in Ceph
- Ceph Client Zone
- Storage Access Zone
  - public_network in Ceph
- Ceph Cluster zone
  - cluster_network in Ceph

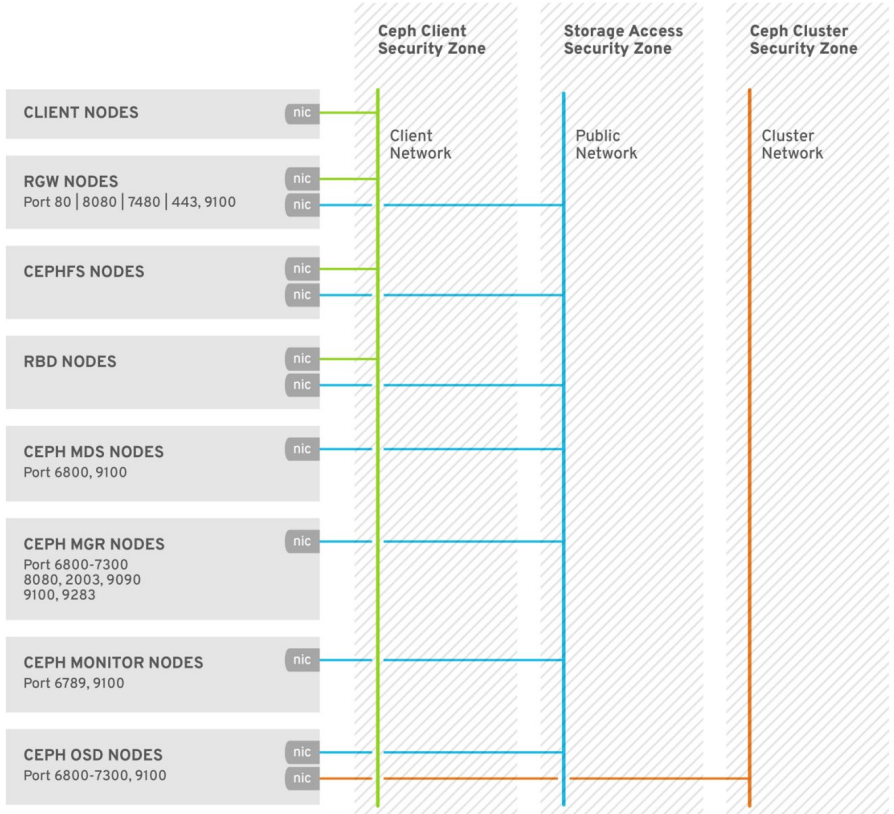
CONNECTING SECURITY ZONES

- Public Zone
  - **not** the public_network in Ceph
- Ceph Client Zone
- Storage Access Zone
  - public_network in Ceph
- Ceph Cluster zone
  - cluster_network in Ceph



CONNECTING SECURITY ZONES

- Public Zone
  - **not** the public_network in Ceph
- Ceph Client Zone
- Storage Access Zone
  - public_network in Ceph
- Ceph Cluster zone
  - cluster_network in Ceph



PRODUCT SECURITY

- Product Security at IBM performs SDL activities across the life-cycle of each release
  - Goal to reduce risk and improve the security of Ceph and Rook
- Suggested improvements & pentesting!
- Many new things coming with collaboration!

IBM PSIRT

- As per usual, we will continue to manifest and document open source materials involved in Ceph
- We'll be onboarding with IBM PSIRT to use new tooling to find even more vulnerabilities than before, and aim to fix even more
  - Expect future updates with details :)

IBM PSIRT

- We will continue to review existing vulns, and release regular security updates, and improve code security preemptively
- We will eventually follow IBM standards to fix vulns and ensure compliance
  - These are often more extensive than Red Hat's
  - Optimistically, we'll have a more secure Ceph

PRODUCT SECURITY INCIDENT RESPONSE

- Responds to incidents and all reported vulnerabilities
- Triages to release fixes in a timely manner
- Works with upstream to review flaws
- Reviews all minor releases, coordinates publication dates with engineering and upstream
- Coordinates release when exploits, not just vulnerabilities, are discovered

PRODUCT SECURITY ARCHITECT

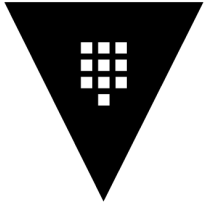
- Ensures correct manifesting, we know (all of) what is in our code and where it is used
- Reviews major releases, approves any exceptions to security policy
- Works with compliance team to validate security policy, federal standards
- Works to suggest design improvements prior to release
  - Cryptographic algorithms, in particular, improvements to msgr2
  - Design choices, key storage improvements, RGW design

NEW WORK AT IBM

- All of the CVE fixes will be continue to be ported to upstream Ceph. We aim to keep all versions of ceph, be that Red Hat, IBM or Upstream, equally secure
- New collaboration produces new challenges, and lots of goals. Fingers crossed
- Reach out to talk about what you want to see with our collaboration with IBM, and feel free to discuss any concerns with us

ENCRYPTION AND KEY MANAGEMENT

- Data at rest (OSD)
  - OSDs can be encrypted with dmcrypt at creation.
  - Write-ahead logs, journals and metadata stores can also be secured
  - LUKS provides a variety of cryptographic options
  - All data at rest is encrypted irrespective of access protocol
  - FIPS 140-2 certified cyphers can be used
- Encryption keys
  - Stored in the Monitor daemon (MON) – or KMS, or K8s Secrets
- Object Gateway (RGW)
  - Data is encrypted at rest relying on OSD strategy
  - Alternatively, data can be encrypted at ingestion with locally managed keys
  - Keys can be managed externally with HashiCorp Vault KMS
  - OpenStack Barbican and KMIP-compatible KMS support is also available



LUKS

ENCRYPTION IN TRANSIT

- Data in transit
  - Ceph's internal protocol can be encrypted as a Messenger v.2.1 protocol option
  - Legacy cleartext protocol is still default for compatibility reasons
  - All data at rest is encrypted irrespective of access protocol
  - FIPS 140-2 certified cyphers can be used
- Client and public security zones
  - TLS security can be used from Object Gateway to S3 clients.
  - TLS termination at HAproxy a special case
- Network hygiene
  - Firewalld at individual nodes

ENCRYPTION IN TRANSIT

- Data in transit
  - Ceph's internal protocol can be encrypted as a Messenger v.2.1 protocol option
  - Legacy cleartext protocol is still default for compatibility reasons
  - All data at rest is encrypted irrespective of access protocol
  - FIPS 140-2 certified cyphers can be used
- Client and public security zones
  - TLS security should be be used from Object Gateway to S3 clients.
  - TLS termination at HAproxy a special case
- Network hygiene
  - Firewalld at individual nodes

ROOK SPECIFIC

- CRDs can be used to encode security preferences
  - Example: client configuration
  - Example: RGW certificate
  - Allows principle of least privilege to easily be implemented
- Rook provides at-rest data encryption as discussed
  - Setup of Msgr v.2 in-flight encryption exists as of 1.9
  - Use software-defined cloud network fabric to segregate traffic
- Standard k8s user permissions apply to persistent volumes
  - Nothing Rook needs to do here
- CSI driver supports KMS
  - PVs can be encrypted with individual keys, limiting key scope

CONTROL PLANE

- SSH
  - Cephadm, ceph-ansible and other tools
  - User (cephadm or ceph) with password-less root access can be used
  - Access is secured with SSH keys
  - Port 22
- Management Dashboard
  - TLS on port 443 (operator facing (storage access zone)
  - Dashboard access zone often tailored by operators to suit local threat model
  - Option for SAML authorization, Kubernetes native authorization
- Manager (MGR)
  - Ceph protocol on port range 6800-7300 (storage access zone)

IDENTITY AND ACCESS

- Cephx
  - Shared secret keys are in use for authentication
  - Mechanism protects cluster from MITM attacks
  - Authentication and authorization are on by default
    - If user is not supplied, provide client.admin as user, and restricted accordingly
- Object Gateway (RGW)
  - S3 user: access key and secret model, option for bucket policy
  - Swift user: access key and secret model
    - Note that default Swift user is sub-user of S3 user, deleting S3 user will delete the Swift user as well
  - Administrative user: access key and secret with access to administrative API
  - User authentication is stored in Ceph pools
    - Identity is an IAM API, consistent and secure
    - Token based authentication with STS API
  - Can couple with OIDC providers (Keycloak, etc), backed by organizational IdP (FreeIPA) for granular role or attribute access

IDENTITY AND ACCESS

- LDAP and Active Directory users can be used as identity services
  - Secure LDAP is highly recommended

- OpenStack Keystone
  - Ceph supports using OpenStack Keystone to authenticate Object Gateway users

AUDITING

Operator actions

- Stored in /var/log/ceph/ceph.audit.log

For example:

```
2018-08-13 21:50:28.727176 mon.reesi001 mon.0 172.21.2.201:6789/0
2097902 : audit [INF] from='client.348389421 -' entity='client.admin'
cmd=[{"prefix": "osd set", "key": "nodown"}]: dispatch

2018-08-13 21:50:28.872992 mon.reesi001 mon.0 172.21.2.201:6789/0
2097904 : audit [INF] from='client.348389421 -' entity='client.admin'
cmd='[{"prefix": "osd set", "key": "nodown"}]': finished
```


In distributed systems, actions may start on one node (dispatch) and propagate to others (finished)

DATA RETENTION

RADOS

- End users generally do not have the ability to read, write or delete objects directly in a storage pool

Ceph Block Device (RBD), Object Gateway (RGW), Filesystem (MDS)

- Users can create, delete, modify volume images, objects or files
- Deletion destroys corresponding RADOS object in unrecoverable manner
  - RBD pools may provide “trash bin” functionality with spare capacity
  - RGW bucket lifecycle supports versioning. Residual data artefacts may persist in storage medium

Secure deletion

- Sanitize retired media by encrypting the OSD contents at rest, and destroying the encryption key

INFRASTRUCTURE HARDENING

- SELinux
  - Red Hat Ceph storage clusters default to SELinux in enforcing mode
- FIPS 140-2 support
  - Certified cryptography can be imported in RHEL “FIPS mode” setup
  - RHEL 8.2 is the most recent certified version
- Hardened binaries
  - `-D_FORTIFY_SOURCE=2`
  - `-D_GLIBCXX_ASSERTIONS`
  - `-fstack-protector-strong`
  - `-fcf-protection`
  - `-fstack-clash-protection`
- Additional Kernel or OS-supplied hardening
  - `SECCOMP`
  - `PIE`
  - `RELRO`
  - `BIND_NOW`
  - `ASLR (all varieties)`


Thank you!

RESOURCES

- Managing and Securing Kubernetes Secrets
  - Rani Osnat - Aquia Security
- Hacking Kubernetes — chapter 6: Storage
  - Andrew Martin and Michael Hausenblas (O'Reilly)
- Data Security and Hardening Guide
  - Red Hat Ceph Storage 5 documentation
- Encrypting Secret Data at Rest
  - Kubernetes documentation
- Recommended Compiler and Linker flags for GCC
  - Survey of Kernel and Userspace hardening options

CREDITS

Federico Lucifredi
Sage McTaggart
Michael Hackett
John Wilkins
J.C. Lopez
Travis Nielsen
Sébastien Han
Ken Dreyer
Kyle Bader