Container Deployment and Security Best Practices

How organizations are leveraging OpenShift, Quay, and Twistlock to deploy, manage, and secure a cloud native environment.

John Morello  
CTO Twistlock

Dirk Herrmann  
OpenShift PM
Who We Are

John Morello  
CTO, Twistlock

Dirk Herrmann  
Openshift & Quay PM
What We Will Cover Today

- The what / why of NIST Special Publications
- SP800-190 and why it matters if you have containers deployed
- How Red Hat and Twistlock can help you achieve the SP800-190 recommendations
- Takeaways & Q + A
The What and Why of NIST
What is a NIST Special Publication?

- Created for significant advancements in technology
- Vendor-agnostic, high level recommendations
- Designed for government and private sector use

The foundation of a collaborative effort between Red Hat and Twistlock to address items inside this publication for our customers.
NIST SP 800-190:
Why This Matters If You Have Containers Deployed
The Challenges of Securing Containers

Why Create a new NIST SP?

- Containers have altered how applications are built, packaged and deployed.
  - Containers scale up and down far more significantly
  - Development cycles have shrunk from months to days
  - Automation has shifted security across the SDLC
- Traditional security guidance does not account for this scale, speed and automation
“Many organizations struggle with the burden of managing security across hundreds of VMs. As container-centric architectures become the norm and these organizations are responsible for thousands or tens of thousands of containers, their security practices should emphasize automation and efficiency to keep up.”

Source: NIST Special Publication 800-190 - Application Container Security Guide
SP 800-190 Looks at 5 Risk Areas

- Image
- Registry
- Orchestrator
- Container
- Host OS
Image Countermeasures

- Use container-specific technology for vulnerability, compliance and secrets management
- Integrate checks and monitoring across the image lifecycle
- Automated, policy-driven enforcement
- Mitigate risks with trusted images
Registry Countermeasures

- Configure development tools, orchestrators, runtimes to require encryption when connecting to registries
- Identify and prevent use of stale images within registries
- Require authentication for read/write access to registries
Orchestrator Countermeasures

- Use least privilege access model
- Separate network traffic by sensitivity level
- Isolate workloads by sensitivity levels
- Build resilience for each node
Container Countermeasures

- Automatically monitor for and remediate CVEs in the container runtime
- Dynamically generate network filtering policy to control inter-container traffic
- Automate compliance checks and leverage technologies like SELinux and seccomp profiles
- Use behavioral learning to profile applications and prevent anomalies
- Full audit trail for container deployment and activity
Host Countermeasures

- Use a container-specific OS when possible
- Do not mix containerized / non-containerized workloads on a host
- Automate vulnerability detection for host OS and its components
- Leverage orchestrators when possible to distribute container jobs across hosts – monitor access patterns
- Run containers with a minimal set of file system permissions
How Redhat and Twistlock help you achieve the SP 800-190 recommendations
Container Technology Architecture

The five tiers of the container technology architecture according to NIST 800-190

Developer Systems → Testing & Accreditation → Registry → Orchestrator → Hosts

- Self-service Provisioning
- Consistent Environments
- Automated build & deploy

Vuln Scanning
Image Signing

Whitelists
RBAC / Auditing

CI/CD pipelines

Secure by default
Network traffic control
Multi-tenant
Workload separation

Container Host OS
Over-the-air Updates
RHEL hardening tools & docs
Container Host OS

What Red Hat and Twistlock offer

Use container-specific host OSs instead of general-purpose ones to reduce attack surfaces.

- Red Hat CoreOS as container-specific host OS
- Immutable host, delivered with OpenShift
- Minimal, secure OS designed to only run containers and one-touch provisioning
- RHEL as an alternative host OS (including all its security features and tooling around)
- Twistlock helps securing the hosts as well (vulnerability & compliance mgt, runtime def)
Container Host OS

What Red Hat and Twistlock offer

However, it is important to note that container-specific host OSs will still have vulnerabilities over time that require remediation.

- Automated updates and CVE remediation (over-the-air-updates)
- Centrally managed or via cluster console
- Updates delivered via OCI images and could be secured as all other workload images
Image Specific Countermeasures
What Red Hat and Twistlock offer

- Use container-specific technology for vulnerability, compliance and secrets mgt
- Integrate checks and monitoring across the image lifecycle
- Automated, policy-driven enforcement

Continuous vulnerability scanning (Clair)
Multiple metadata sources for Red Hat and other content
Scalability up to millions of images battle tested over years (Quay.io)
Notifications and Reporting / Dashboards*
Policy enforcement in OpenShift based on various attestations (incl. vulnerabilities)*

* coming soon
Vulnerability Scanners

What Red Hat and Twistlock offer

Detects vulnerabilities and malware in containers and hosts at every stage of the lifecycle for Linux and Windows (4.1.1, 4.4.1, 4.5.3, 4.2.2, 4.1.3)

Environmental context and other risk factors are used to generate a custom risk score for each CVE

30+ upstream sources are combined for the lowest possible false positive rate

Risk tree + per-layer views help pinpoint patching efforts
Registry Countermeasures
How Red Hat Quay, OpenShift and Twistlock address them

- Host can only connect to the registry over encrypted channels
- All access requires authentication, especially write access (push)
- All write access is audited and read actions are logged
- Auto-pruning of (outdated / vulnerable) stale images*
- Context-aware authorization control to actions
- Integrated automated scanning of images

* implementation not based on pruning but on restricting their usage via policies, coming soon
Image Countermeasures - Untrusted Images

- Capability to centrally control exactly what images and registries are trusted in their environment
- Discrete identification of each image by cryptographic signature
- Enforcement that all hosts only run images from these approved lists
- Validation of image signatures before image execution to ensure images are from trusted sources and have not been tampered with
- Ongoing monitoring and maintenance of these repositories to ensure images within them are maintained and updated as vulnerabilities and configuration requirements change
At Build.....What Are The Considerations?

What Red Hat and Twistlock offer

Red Hat Container Catalog & Health Index
Red Hat Quay as the content ingress point for all containerized applications
Only import trusted images (whitelists)
Automated vuln’ scanning after push attestations for policy mgt / enforcement
Build Pipeline part of OpenShift
Webhooks, triggers and notifications

Centrally define trusted image and enforcement policies to block non-trusted deployments (4.1.5,4.2.1)
Full-lifecycle compliance checking for misconfigurations, unencrypted secrets, and industry best practices (PCI, GDPR, etc) (4.1.2,4.1.4)
I’m Moving Code to Production...Now What?

What Red Hat and Twistlock offer

Several technologies and tools to harden environments and clusters

Environment specific policies to ensure highest level of protection for production*

Policy enforcement based on various attestations (signatures, vuln’s, labels)*

Continuous vulnerability scanning and notifications (including rebuild triggers)

Automated operations across the stack

Automatically enforced whitelist-based models to prevent anomalous behavior across containers and hosts (4.4.2, 4.4.4, 4.4.5, 4.5.1)

East west firewallsing that uses machine learning to microsegment your environment (4.4.2)

Running containers and hosts are continually re-assessed for new vulnerabilities (4.4.1, 4.5.3)

* coming soon
Container-Aware Runtime Defense Tools

What Red Hat and Twistlock offer

Automatically enforced whitelist-based models (process, network, filesystem, system calls) to prevent anomalous behavior across containers and hosts (4.4.2, 4.4.4, 4.4.5, 4.5.1)

East west firewalling that uses machine learning to microsegment your environment and container-aware L7 firewalls (4.4.2)

Running containers and hosts are continually re-assessed for new vulnerabilities (4.4.1, 4.5.3)
NIST 800-190 Key Take-Aways

- Adapt your IT organization and operational model to reflect new paradigms
- Use container host operating system variants for smaller attack surface
- Keep workloads separated by sensitivity levels
- Use tools and processes built for the new paradigms and technologies
- Carefully select content and implement a content governance process
- Choose tools that give you visibility into your full stack - containers, hosts and orchestration.
Takeaways + Q&A
THANK YOU

plus.google.com/+RedHat
linkedin.com/company/red-hat
youtube.com/user/RedHatVideos
facebook.com/redhatinc
twitter.com/RedHat