Trusted CI Webinar Series

Today’s webinar topic is “Best Practices for Academic Cloud Service Providers” with Rion Dooley. Our host is Jeannette Dopheide.

The meeting will begin shortly. Participants are muted. Click the Chat button to open the chat view and ask a question.

This meeting will be recorded.

The Trusted CI Webinar Series is supported by National Science Foundation grant #1547272.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the NSF.
SECURITY BEST PRACTICES FOR ACADEMIC CLOUD SERVICE PROVIDERS

STAKEHOLDERS

- Cloud resource providers
- Cloud service providers
- Academic cyberinfrastructure providers
- Users
SCOPE

- IaaS
- PaaS
- Saas
SCOPE

- Gateways, web apps, web services, etc
- Orchestration, db, queues, registries, config mgmt.
- Storage, compute, load balancing, provisioning, networking, IAM

- Hypervisor, cloud management
- Rack and stack
- Power and cooling
PARTICIPANTS
DOWNLOAD THE OFFICIAL WHITE PAPER

http://hdl.handle.net/2022/22123
TERMS OF ENGAGEMENT

1. Security is a shared concern between a cloud service provider and a cloud service user, neither can expect the other to fully address security.

2. A clean delineation between cloud service provider and cloud service user of security responsibilities is critical to ensure all responsibilities are met.

3. The cloud service provider has the responsibility to ensure all security responsibilities are articulated and the cloud service user is educated about how to fulfill their responsibilities.
TERMINOLOGY

- **Cloud Service**: An internet accessible compute service with the intended purpose of allowing users to instantiate and manage system images.
- **Image**: A data file containing the contents of a virtual machine or container.
- **Resource/Service Provider**: A operator of a cloud service.
- **Running image**: An instantiated image that is actively processing.
- **Security concern**: Any issue that increases security risk to the cloud service.
- **User**: Someone instantiating and managing running images.
THE RECOMMENDATIONS

► Disseminate Localized Best Practices
► Ensure Image Trustworthiness
► Provide Method to Manage User Secrets
► Support Privileged Access within Images
► Empower Users with Self-service DNS Management
► Provide Method to Manage User Configurations
► Provide Service Accounts
► Offer Monitoring Services
► Offer Identity and Access Management-aware Continuous Integration / Continuous Delivery Services
DOWNLOAD THE OFFICIAL WHITE PAPER

http://hdl.handle.net/2022/22123
**Use Case**: Users of a cloud service will have a number of tasks that are common across cloud services, but vary in details whose implementation can lead to errors with security consequences.

- How to carry out non-interactive tasks
- Identifying images that are ready for production versus those that are not.
- The configuration of IT automation tools (Ansible, chef, puppet, terraform, etc) for the cloud service’s specific environment.
- What minimal custom configuration of images is needed to make them secure.
- How should development environments securely interact with the cloud service to deploy, debug, and manage service?
DISSEMINATE LOCALIZED BEST PRACTICES

Security Concern:

▶ Each cloud implementation has its own nuances.
▶ Without clarification and guidance, users are more likely to make mistakes.
   ▶ Hard-coding secrets and sensitive information.
   ▶ Unpatched images
   ▶ Failing to notice anomalous behavior
DISSEMINATE LOCALIZED BEST PRACTICES

Recommendations

▶ Provide best practice documents and support services regarding the issues identified in this use case.
▶ Detect common patterns of misbehavior so that users can be alerted and educated.
DISSEMINATE LOCALIZED BEST PRACTICES

Examples

- Wiki, blog, Google doc, webinars
- Sample repositories, online training
- “Easy button” examples
- Reference implementations, prebuilt images, published orchestrations
- Detect common patterns of misbehavior so that users can be alerted and educated.
ENSURE IMAGE TRUSTWORTHINESS

Use Case: In the process of their research, users may execute images of unknown provenance on a cloud resource.

- Prebuilt software stacks
- Aging tutorial examples
- Untrusted container images
- Published automated deployments
ENSURE IMAGE TRUSTWORTHINESS

Security Concern:

▶ How are curated in a way that creates trust?
▶ How are images secured and applications contained?
▶ How/when are updates applied and communicated?
ENSURE IMAGE TRUSTWORTHINESS

Recommendations: Provide a managed, hosted image registry.

- Digital signatures
- Vulnerability and secret scanning (auto invalidation & alerts)
- Robust access control
- SSO authentication
- API for automation
- Web UI for management, viewing diffs, history, etc.
- Tagging and versioning
ENSURE IMAGE TRUSTWORTHINESS

Example Virtual Machines

- OpenStack’s Image Service
- VMware vCenter Server
- JFrog Artifactory

Example Container Registries

- Docker Trusted Registry
- Twistlock
- Quay
- Black Duck OpsSight
- Gitlab Container Registry
- JFrog Artifactory
- Nexus
Provide Method to Manage User Secrets

**Use Case**: Users need a reliable solution for managing secrets such as API keys, signing certificates, passphrases, and other sensitive information used to manage their environment.

- Management, provisioning, storage, and retrieval
- Distinct from authentication
- Not restricted to any particular type of secret
Provide Method to Manage User Secrets

Security Concern:

- Cloud environments facilitate service scaling and automation.
- Storage of secrets such as API keys or passphrases is frequently required.
- In lieu of a better, recommended way, users will do the easiest thing for them.
- Easy == next to nothing
- This can be avoided by providing strong and tractable secret management within their infrastructure.
Provide Method to Manage User Secrets

Recommendations: Service providers should offer a hosted secret manager.

- API driven
- Mountable on the file system
- Support for webhooks and event notifications
- Access history available to query per stored secret.
- Bidirectionally encryption
- Optional hashing challenge for authentication checks
- Integrated authorization, including OTP and delegation
- SSO authentication
Provide Method to Manage User Secrets

Examples

- Keywhiz (https://square.github.io/keywhiz/)
- Barbican (https://github.com/openstack/barbican)
- AWS key management service (https://aws.amazon.com/kms/)
- AWS Secrets Manager (https://aws.amazon.com/secrets-manager/)
Support Privileged Access within Images

**Use Case:** Users of cloud services need privileged access within their running image to perform their desired tasks, e.g., running services that require registered network ports.
Security Concern:

- Privileged access for a user means no operational constraints for the user.
- Any processes put into place by the operator can be circumvented or disabled by a privileged user.
- This increases risk to the service operator, other users, and third parties.
Support Privileged Access within Images

Recommendations:
- Limit image functionality via the network
- Network monitoring, passive or active

Anti-patterns:
- Encourage users to run security-enhanced O/S
- Disallow the granting of privileged access.
- YOLO
Empower Users with Self-service DNS Management

**Use Case**: Users need the ability to enable secure communications across their applications, infrastructure, and third-party services.

- Applications often need to communicate.
- Communication may contain sensitive information from user and/or raw data.
- Securing communication is critical step in system security.
- SSL/TLS certs require money or TLD control
- Users cannot and will not pay to maintain valid ssl certs over time.
- Facilitating the acquisition and use of valid, secure certs for users helps everyone.
Empower Users with Self-service DNS Management

Security Concern:

- SSL/TLS is standard for securing communication.
- Trusted certs are available for free from Let’s Encrypt (https://letsencrypt.org/).
- Requires TLD control.
- Painful for users to bootstrap trust with invalid certs.
- Users avoid pain.
Empower Users with Self-service DNS Management

Recommendations: Automatic DNS record generation & DNSaaS:

- Automatically generate A and cname records for new instances.
- Provide a self-service DNS management API.
- Allow subdomain management to users based off auto-generated domains.
Empower Users with Self-service DNS Management

Example DNS Management Components

▶ AtomiaDNS (http://atomiadns.com/)
▶ PowerDNS (https://www.powerdns.com/)

Example DNSaaS Components:

▶ OpenStack’s Designate (https://docs.openstack.org/designate/latest/)
▶ Consul (https://www.consul.io/)
Provide Method to Manage User Configurations

**Use Case:** Users with more than one similar application to manage across images may lack consistency in the implementation of security controls and state between applications.

- Users may have multiple applications they manage.
- Configuration and toolchains may differ between applications.
- Security controls frequently diverge in these situations.
- Inconsistencies can lead to system compromises.
- One compromise can lead to compromise of entire stack.
Provide Method to Manage User Configurations

Security Concern:

- More applications leads to more complexity
- Complexity over time can lead to misconfigurations and missing controls.
- This leaves systems open to a wider range of attacks and increases the security risk of the overall environment.
- One compromise can lead to many.
- Users may not know how to restore that host's configuration to a known good configuration.
- Urgency may lead to not properly re-implementing security controls.
Provide Method to Manage User Configurations

**Recommendations:** Provide configuration managers

▶ Providers should offer solutions that provide some level of management over the applications users run within their images.
▶ Made configurations available when the user needs it:
  ▶ Passed to the image on startup, e.g., atmosphere,
  ▶ Pulled in from the application or other system software running within the image
Provide Method to Manage User Configurations

Examples:

- Ansible (https://www.ansible.com/)
- Puppet (https://puppet.com/)
- Chef (https://www.chef.io/chef/)
- Salt Stack (https://saltstack.com/)
- AWS Config (https://aws.amazon.com/config/)
- Terraform (https://terraform.io/)
- Helm (https://helm.io/)
- Atmosphere's Ansible Instance Deployment Setup3
Provide Service Accounts

**Use Case**: Users should be able to carry out non-interactive tasks without losing full account access if a task or service is compromised.

Example tasks:

- Deploying code
- Patching images
- Rotating hosts
- Refreshing tokens
- Backing up data
- Auto-scaling
- Batch processing
Provide Service Accounts

Security Concern:

- Users frequently use a single account to managing all their cloud utilization.
- Single compromise means all current and future systems were effectively compromised.
- Requires “death sentence” to mitigate risk.
- Creates risk for the user to go all-in with a single cloud provider.
Provide Service Accounts

Recommendations: Service accounts

- Enable the creation and management of self-service service accounts.
- Make service accounts first class identities within the infrastructure.
- Allow for the granting and restriction of specific security contexts these accounts.
- Allow for temporal and programmatic invalidation of service accounts.
- Integrate the service accounts into the existing IAM system to enable interchangeable use with existing user accounts.
Offer Monitoring Services

**Use Case:** Users need to be aware of the current state of their systems including hosts, firewalls, networks, storage services, and infrastructure.
Offer Monitoring Services

Security Concern:

- Comprehensivity is a core principle of good security.
- The average user will not have the time or expertise to implement comprehensive logging across their applications.
- Without visibility, users are at risk.
- Without insight, the risk will remain and breaches will continue.
Offer Monitoring Services

Recommendations:

▶ Leverage existing monitoring solutions already in place by each RP.
▶ Provide interfaces to enabling retrieval of pertinent events regarding the users’ account, image, applications, etc.
▶ Provide multi-channel notifications of infrastructure service outages:
  ▶ Webpage
  ▶ Alert subscriptions (email, webhook, slack, etc)
  ▶ API
Offer Monitoring Services

Example Monitoring Interface Solutions:

- Nagios
- Icinga
- AWS CloudWatch
- Vulnerability scanners such as OpenVAS
- Prometheus
- Grafana

Example Status pages:

- Cachet
- Stashboard
Offer IAM-aware CI/CD Services

**Use Case:** Users will utilize Continuous Integration / Continuous Delivery (CI/CD) services in order to ease development and operations (DevOps) of their applications.

- Jenkins
- Ansible Tower
- Drone
- GitLab Continuous Integration & Deployment
- etc.
Offer IAM-aware CI/CD Services

Security Concern:
Users can easily setup CI/CD solutions.
Sensitive information is managed by the CI/CD server. User is unaware of how.
CI/CD server may not be secured.
Data may not be secured.
Puts the server, host, and user’s infrastructure infrastructure at risk.
Offer IAM-aware CI/CD Services

**Recommendations:** Offer hosted, multi-tenant CI/CD services

- Providers should offer hosted, multi-tenant CI/CD services
- Services should integrate with the existing secret and monitoring solutions
- Should provide hosting and capacity outside the user’s allocation.
- Should leverage best practices for information security within the hosted service.
Offer IAM-aware CI/CD Services

Example CI/CD Services (w/Integrating IAM) Solutions:

- Jenkins ([https://jenkins.io/](https://jenkins.io/))
- Tower ([https://github.com/ansible/awx](https://github.com/ansible/awx))
- TravisCI ([https://travis-ci.org/](https://travis-ci.org/))
- Rundeck ([https://www.rundeck.com/open-source](https://www.rundeck.com/open-source))
- CircleCI ([https://circleci.com/](https://circleci.com/))
- TeamCity ([https://www.jetbrains.com/teamcity/](https://www.jetbrains.com/teamcity/))
DOWNLOAD THE OFFICIAL WHITE PAPER

http://hdl.handle.net/2022/22123
THANK YOU

http://hdl.handle.net/2022/22123
Questions?

Please take our survey
About the Trusted CI Webinar series

To view presentations, join the discuss mailing list, or submit requests to present, visit:

https://trustedci.org/webinars

The next webinar is January 28th at 11am Eastern.
Topic: Research Security Operations Center (ResearchSOC)
Speaker: Von Welch & RSOC leadership team