



KubeCon



CloudNativeCon

North America 2023

Cloud Native Application Threat Modeling and Adversary Emulation: Techniques and Tools

Rafik Harabi - Sysdig

Who Am I?

- Senior Solution Architect at Sysdig, Cloud Security Advocate
- Focus on Cloud Native Security
- Previously working on go to Cloud programmes (Deloitte, NTT Data..)



[rafikharabi](#)



[@rafik8_](#)



Agenda

- Cloud Native Application building blocks
- The multitude of Cloud attack surfaces and it challenges
- Threat modeling technique for Cloud Native Applications
- Adversary Emulation for Cloud Native Applications
- Tooling
- Takeaways

Once, there was a perimeter

You had a perimeter **guarded**
by a firewall

Detecting intrusions was
your breach indicator



Now, there is no perimeter in the cloud



Cloud providers own external connections



Cloud is exposed to the outside world



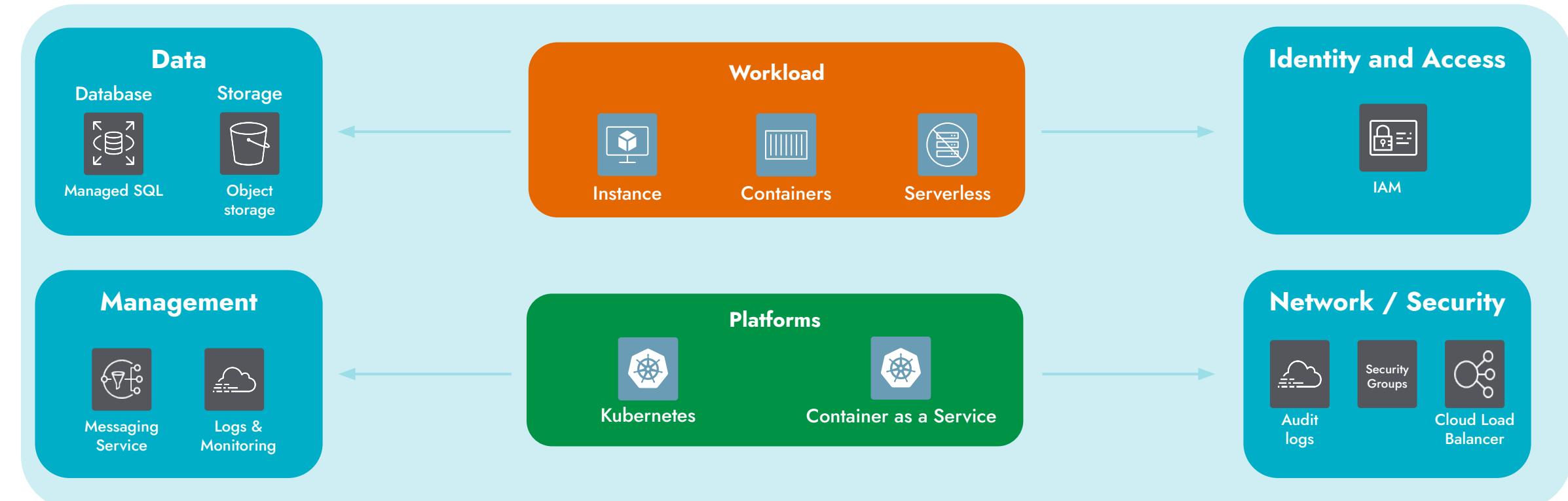
You need to control access to services your team uses



You need to detect unusual activity



Cloud Native Application building blocks



Cloud Infrastructure

Cloud Provider

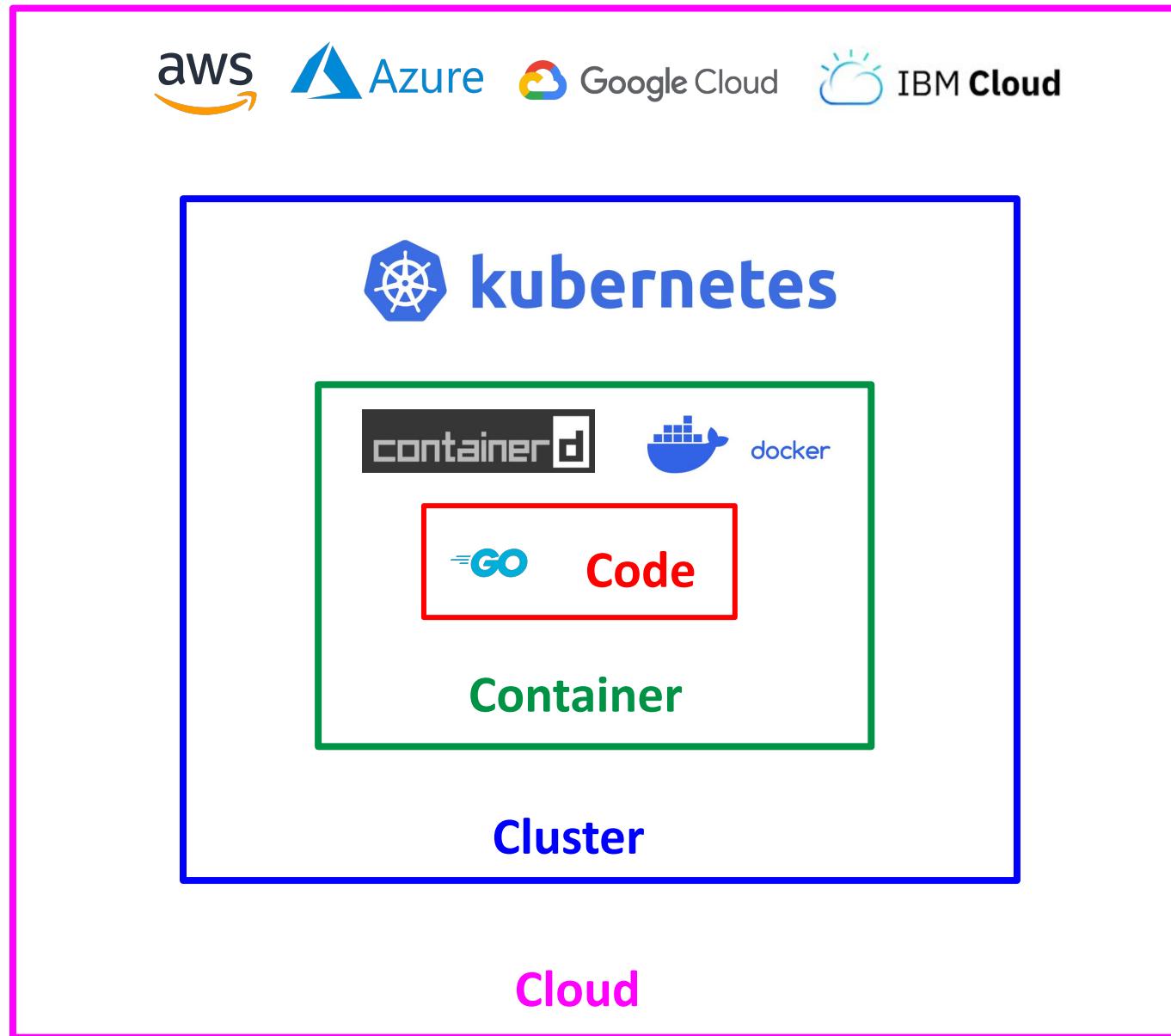


Azure

Google Cloud

IBM Cloud

Cloud Security layers

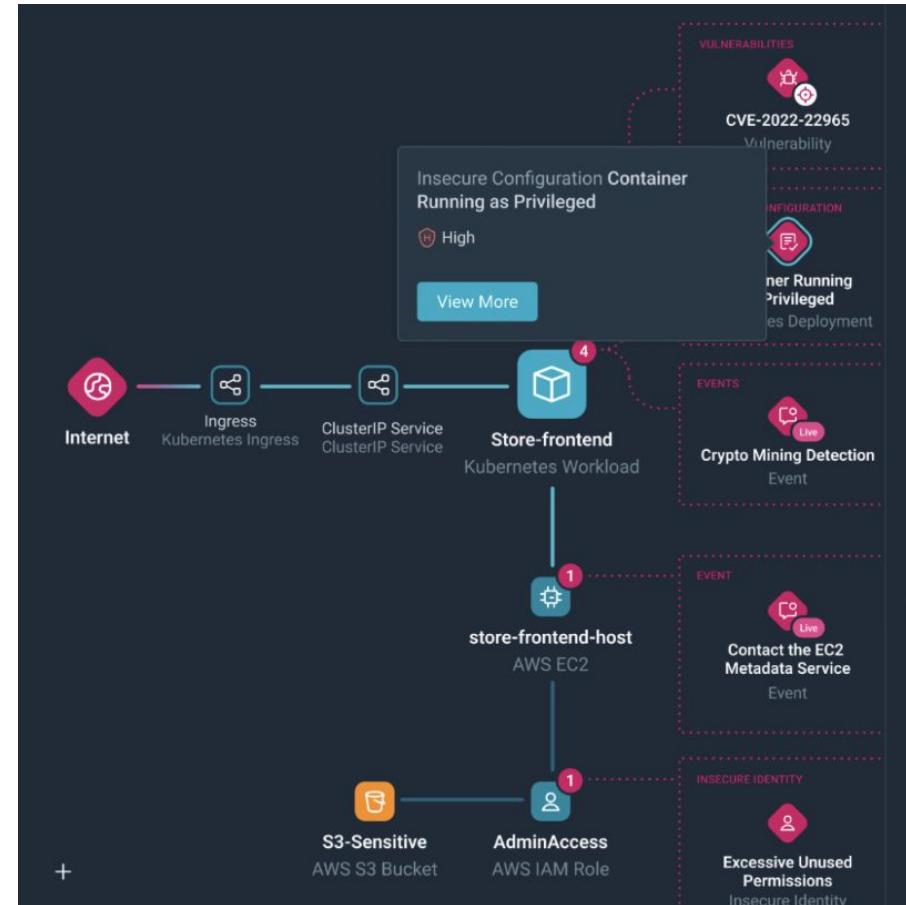
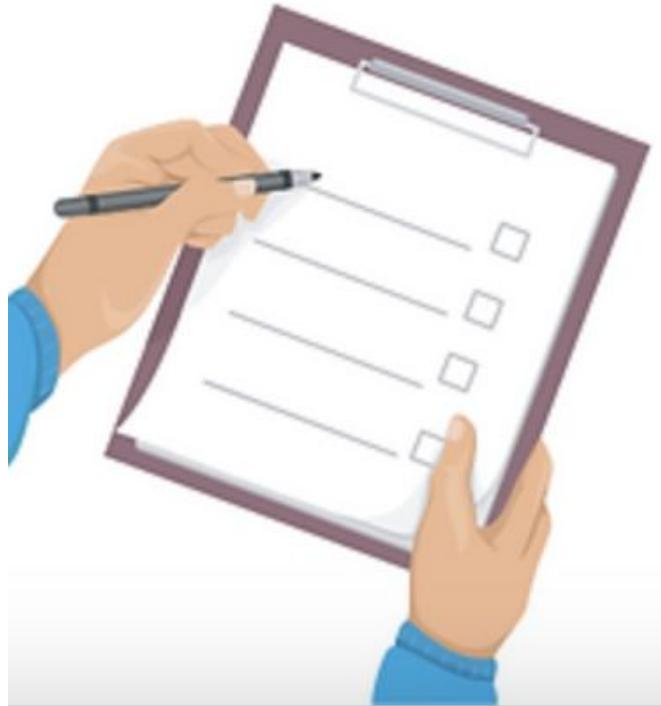


CNA Security Challenges

- Dynamic attack surface,
- Threat actors are using your tools today,
- Distributed systems and microservices enlarge attack surface,
- Number of calls generated by distributed systems,
- Lack of visibility,
- Cloud delivery vs security process speed.

Attacker vs Defender mindset

"Defenders think in lists, attackers think in graphs; as long as this is true, attackers win." John Lambert - Microsoft Security Research



Definitions

Threat Modeling:

“Threat Modeling works to identify, communicate and understand threats and mitigations within the context of protecting something of value.” OWASP

Goal: improving security by identifying threats and provide mitigation.

Adversary Emulation:

“Simulating the tactics, techniques, and procedures (TTPs) employed by real-world threat actors to test an organisation's resilience against different types of attacks.”

Goal: understand how an adversary would attempt to compromise an organization.

Threat Modeling Techniques

Methodologies:

- **STRIDE**: created at Microsoft, defines 6 categories of threats.
- **Attack Tree**: multi-leveled diagrams consisting of one root, leaves, and children. when the root is satisfied, the attack is complete.
- **Dataflow**: graphical representations of your system and should specify each element, their interactions and helpful context.

Pillars:

- Systems Architecture
- Actors
- Threats
- Mitigations

System design and Architecture

Define system components, their interactions and boundaries

Threat Modeling

Explore and listing potential threats.

Threat List

Categorization, Prioritization and Mitigation

STRIDE Approach

Threats

Spoofing

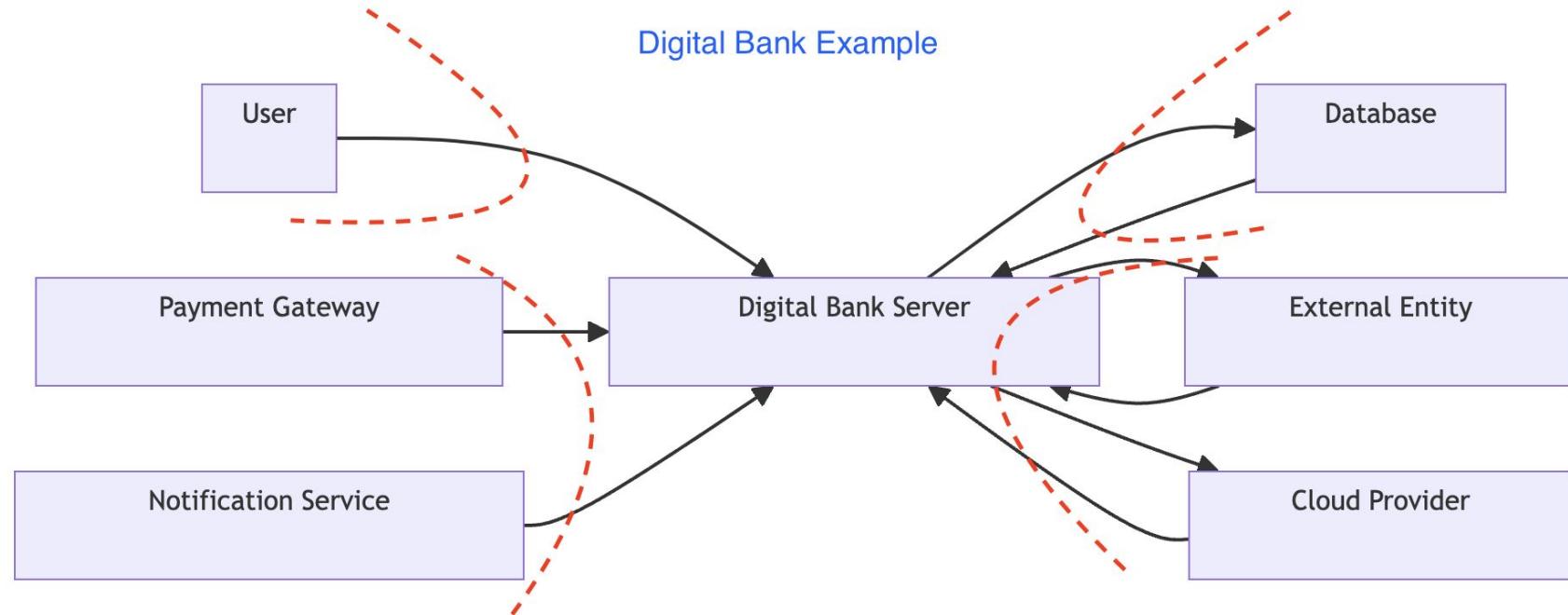
Tampering

Repudiation

Information Disclosure

Denial of Service

Elevation of Privilege



➡ Decompose the system into components, modules and identify relationship between them.

Threat Modeling Personas

Business Owner

- Balances business requirements with the mitigations proposed to address threats

App/Service Developer

- Facilitates design brainstorming implement mitigations

The Adversary

Simulate an unauthorized user to find Threats

The Defender

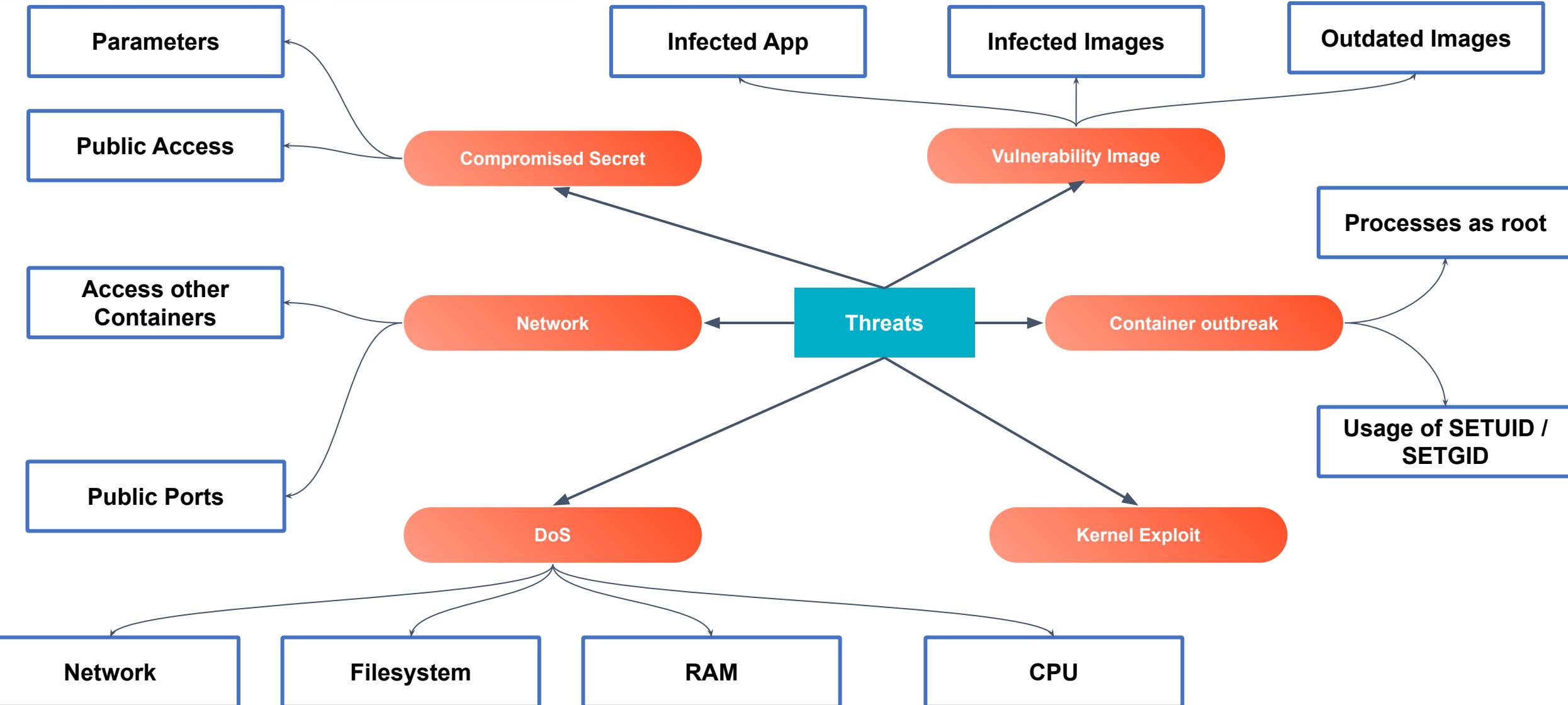
Defines security controls to mitigate the threats

Security Architect

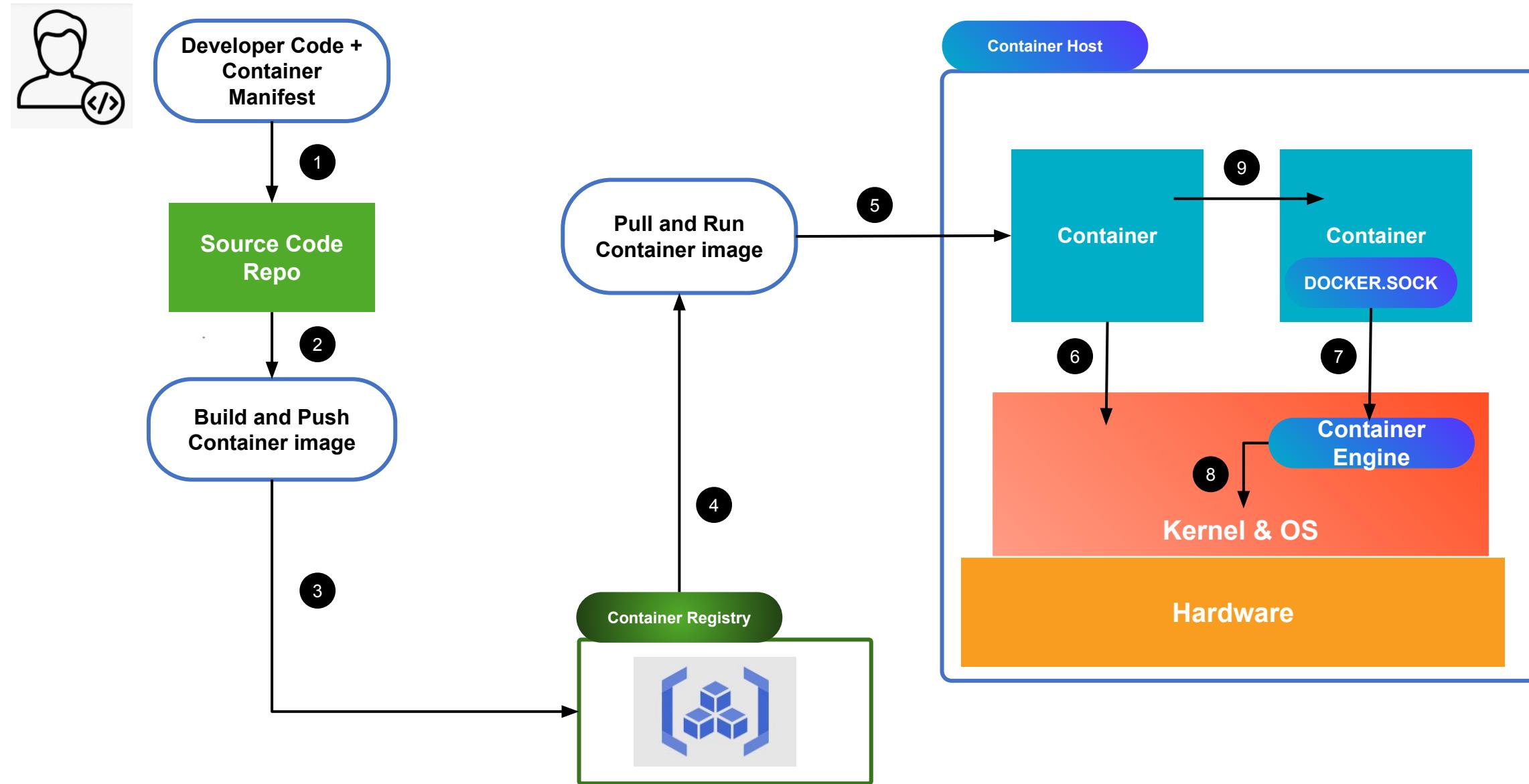
Provides security guidance

Threat Modeling for Cloud Native

Container Threats

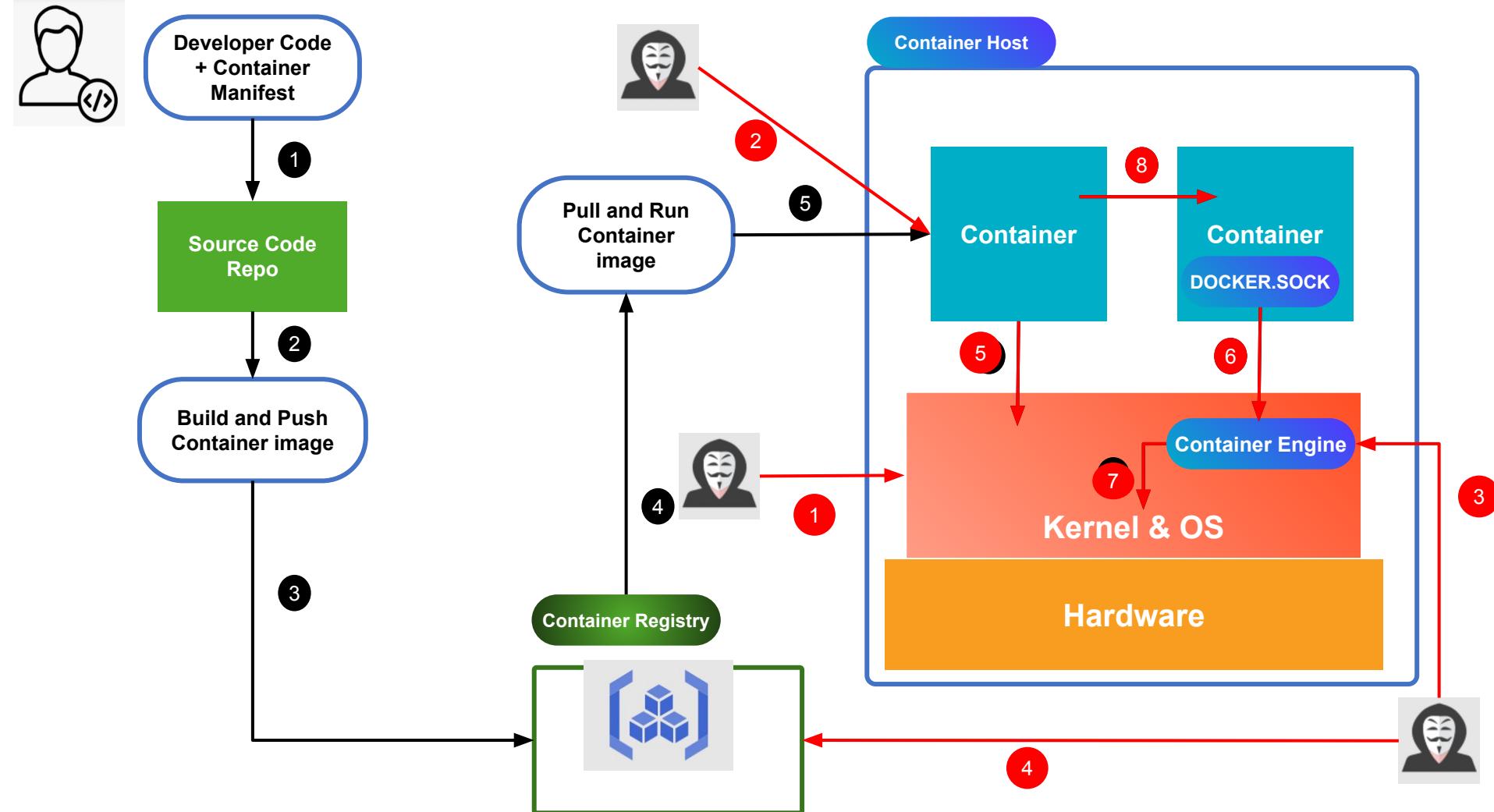


Container Data Flow Diagram

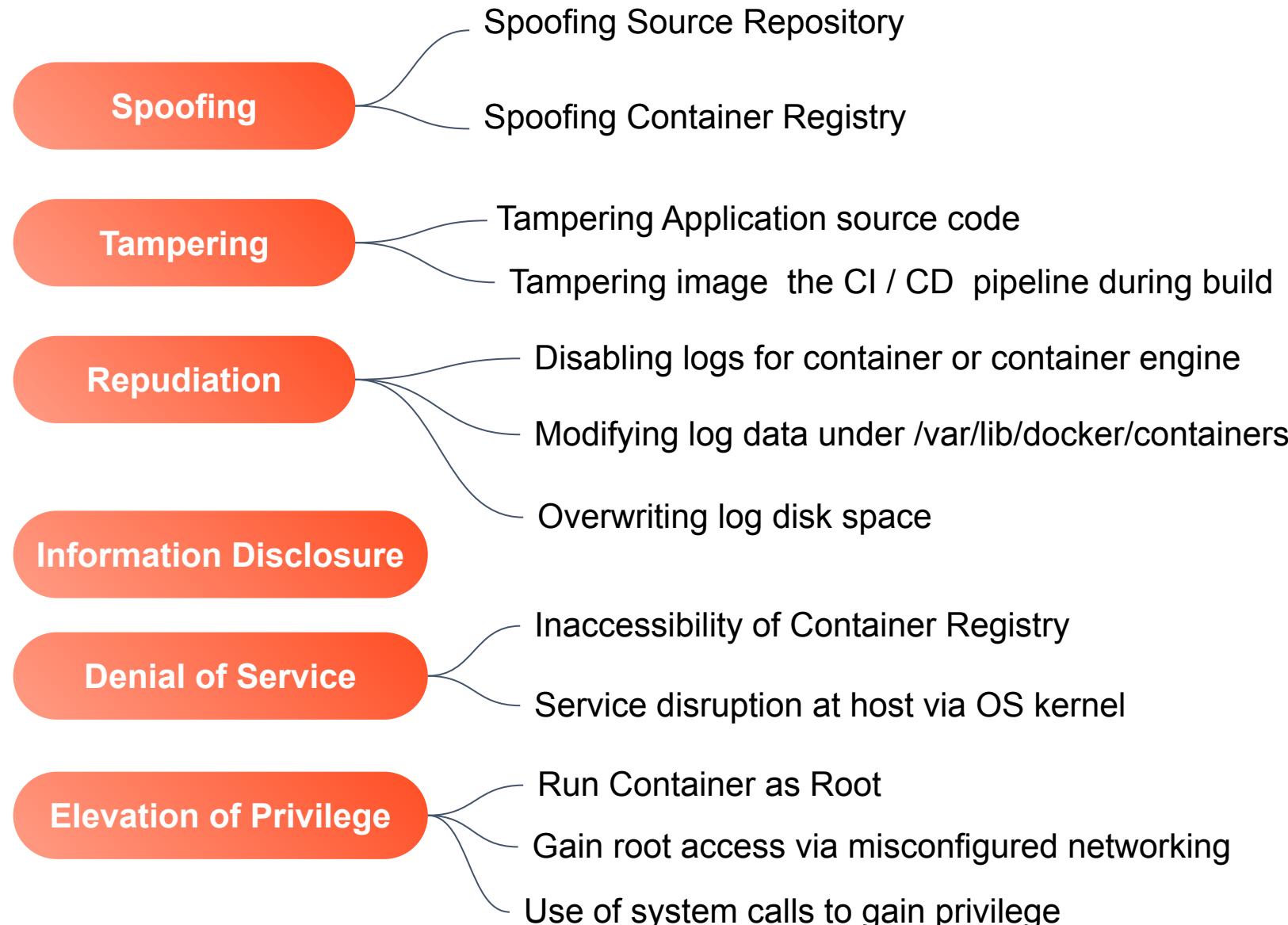


Container Threat Vectors

- 1 Vulnerable OS/Container engine
- 2 Vulnerable application
- 3 Exposed Container engine
- 4 Insecure image registry
- 5 Privileged containers
- 6 Misconfigured container
- 7 Privilege escalation on host
- 8 Insufficient Network isolation

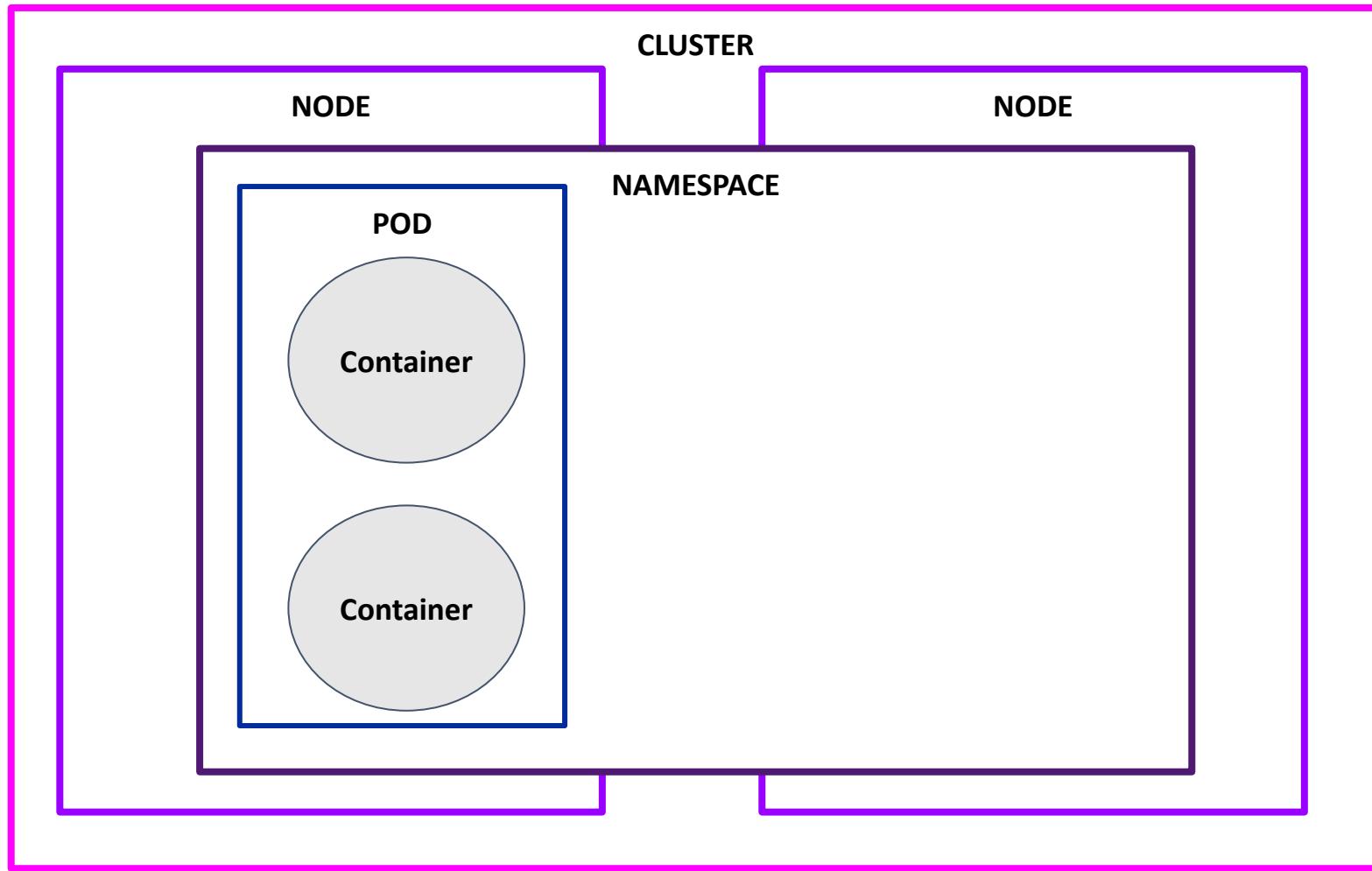


Container Threat Analysis (STRIDE)

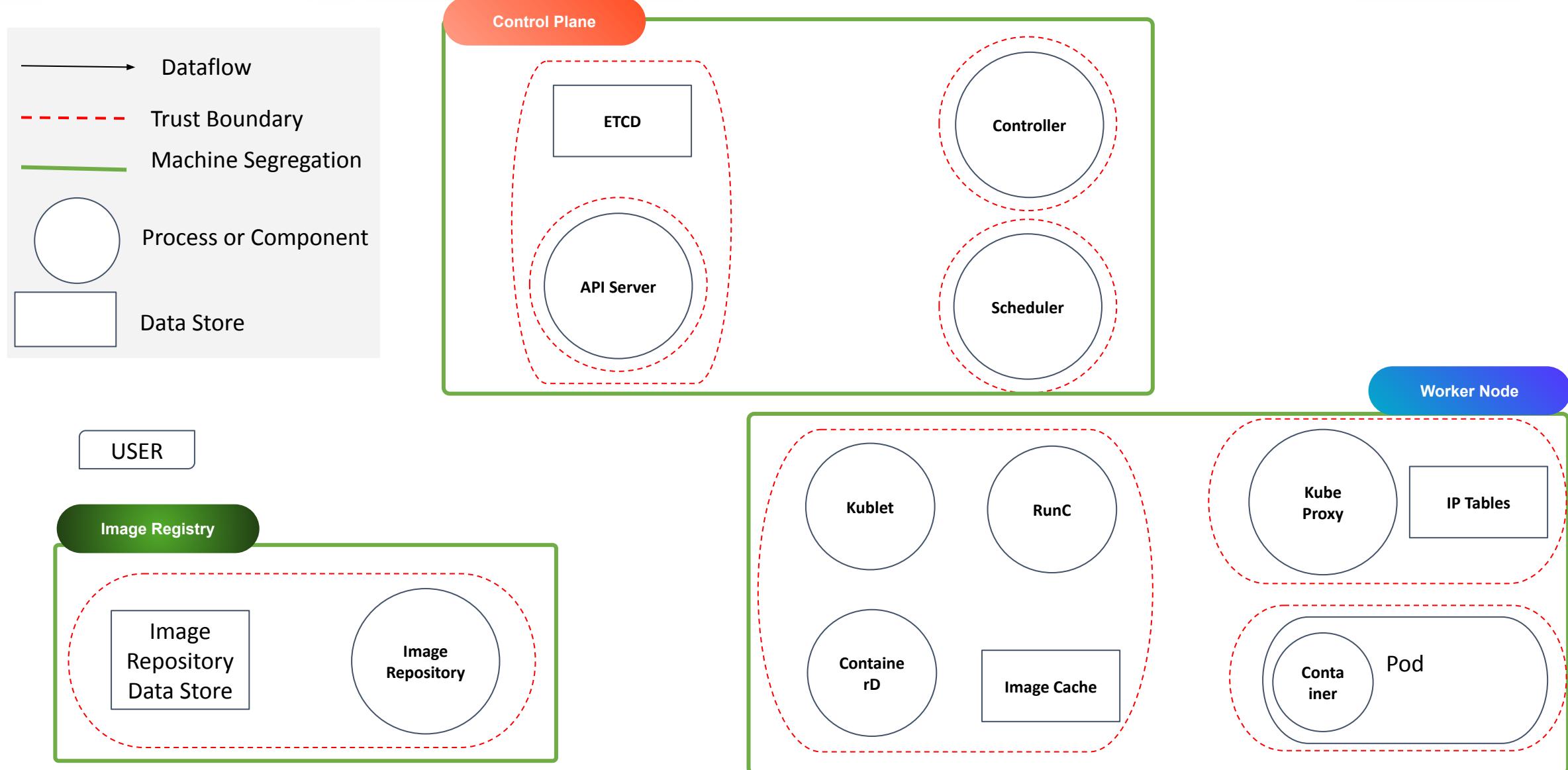


Kubernetes Threat Modeling

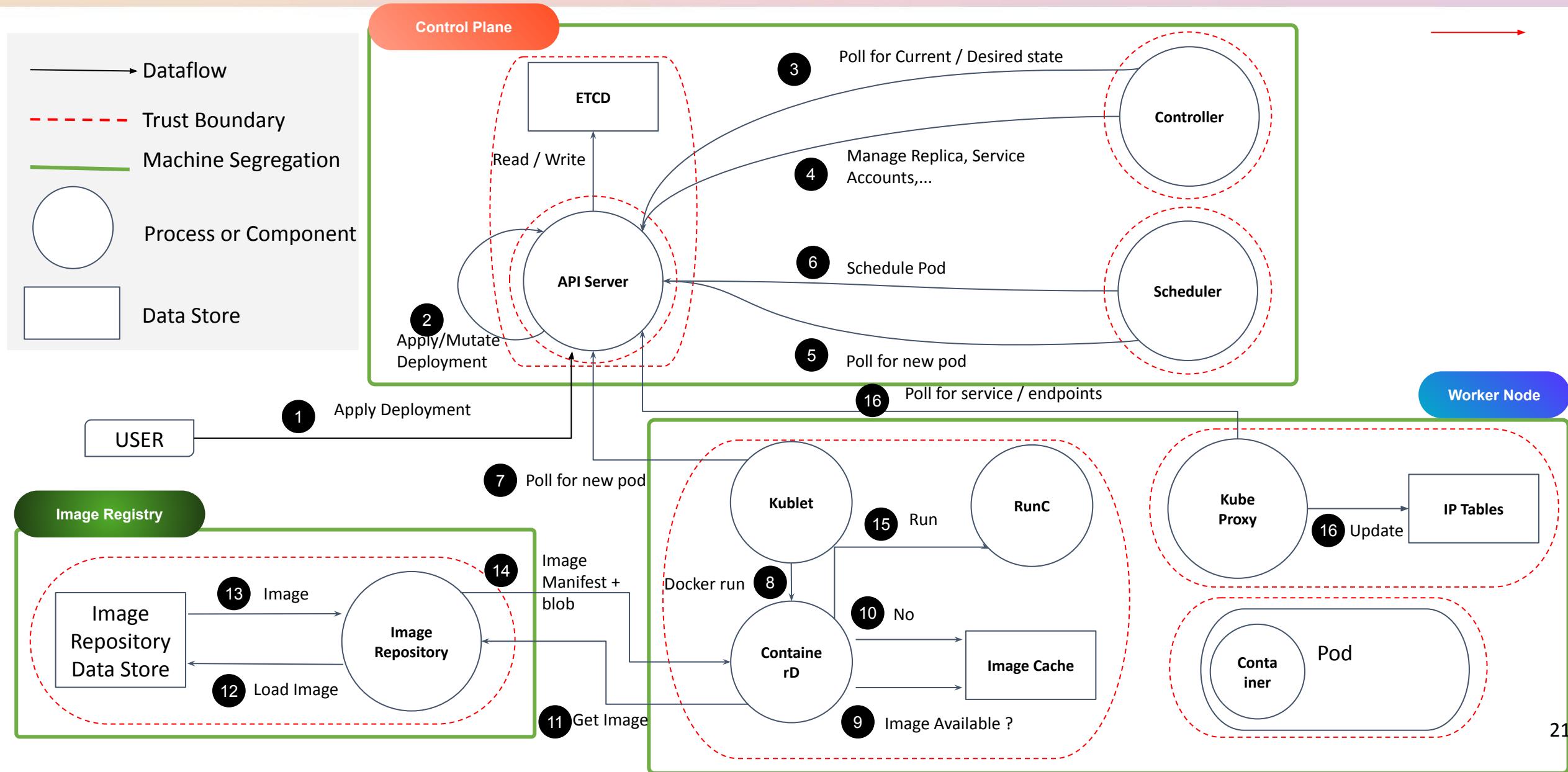
- Kubernetes default Security Boundaries



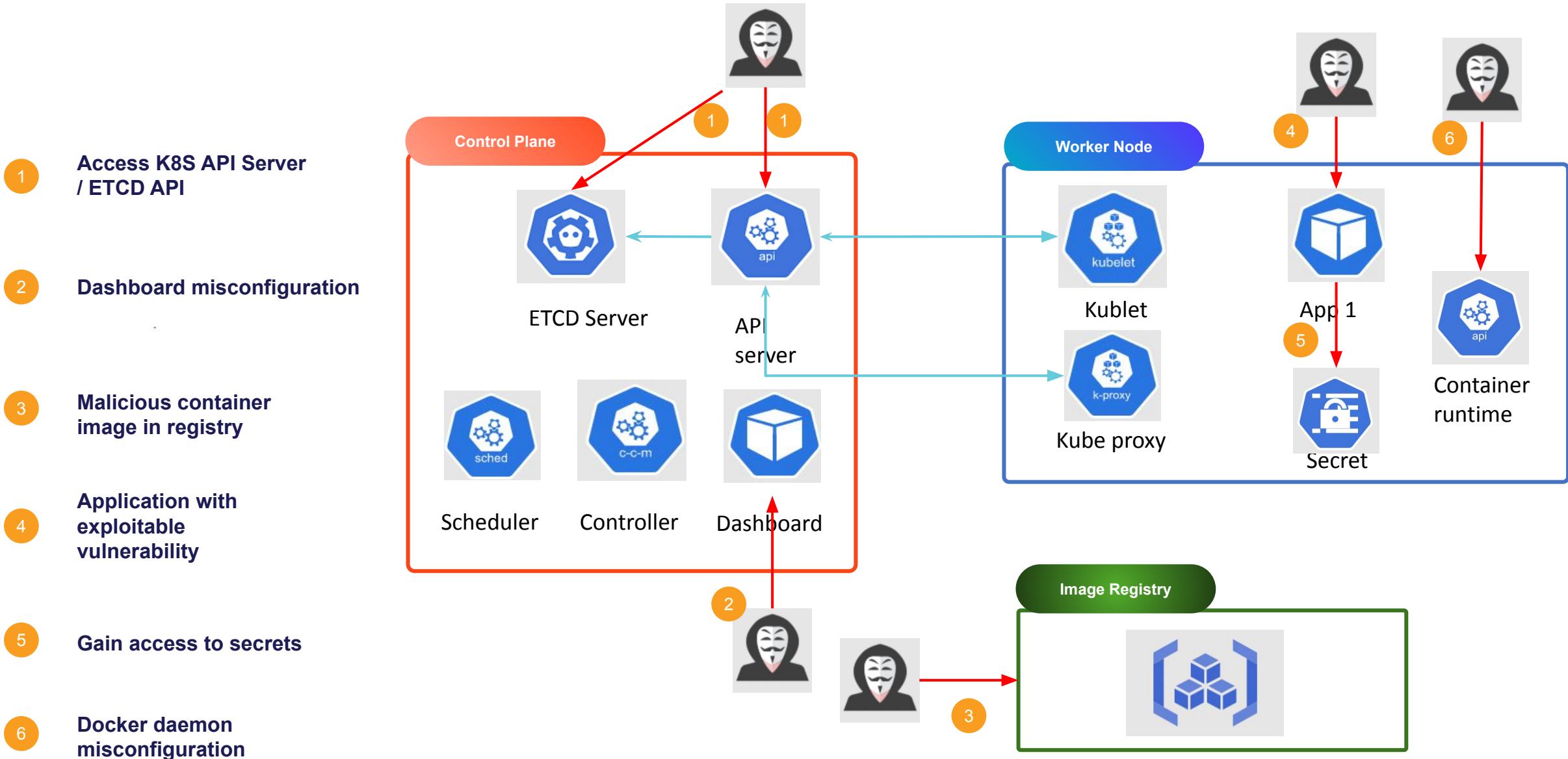
Kubernetes Attack Surface



Kubernetes Data Flow Diagram

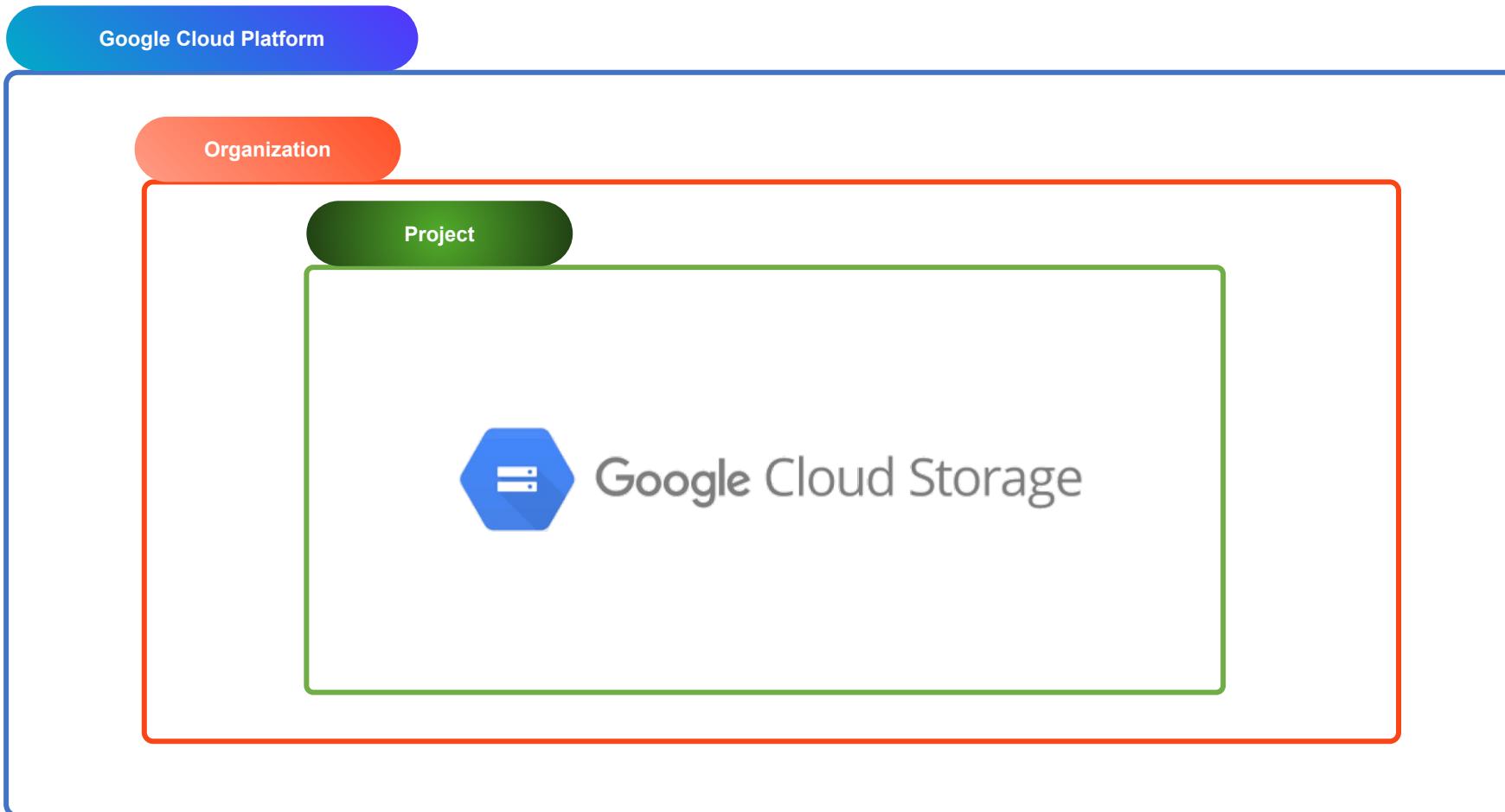


Kubernetes Attack Vectors



Cloud Threat Modeling

- We will be using the same threat modeling STRIDE
- We will take Google Cloud Storage as example



Cloud Threat Modeling

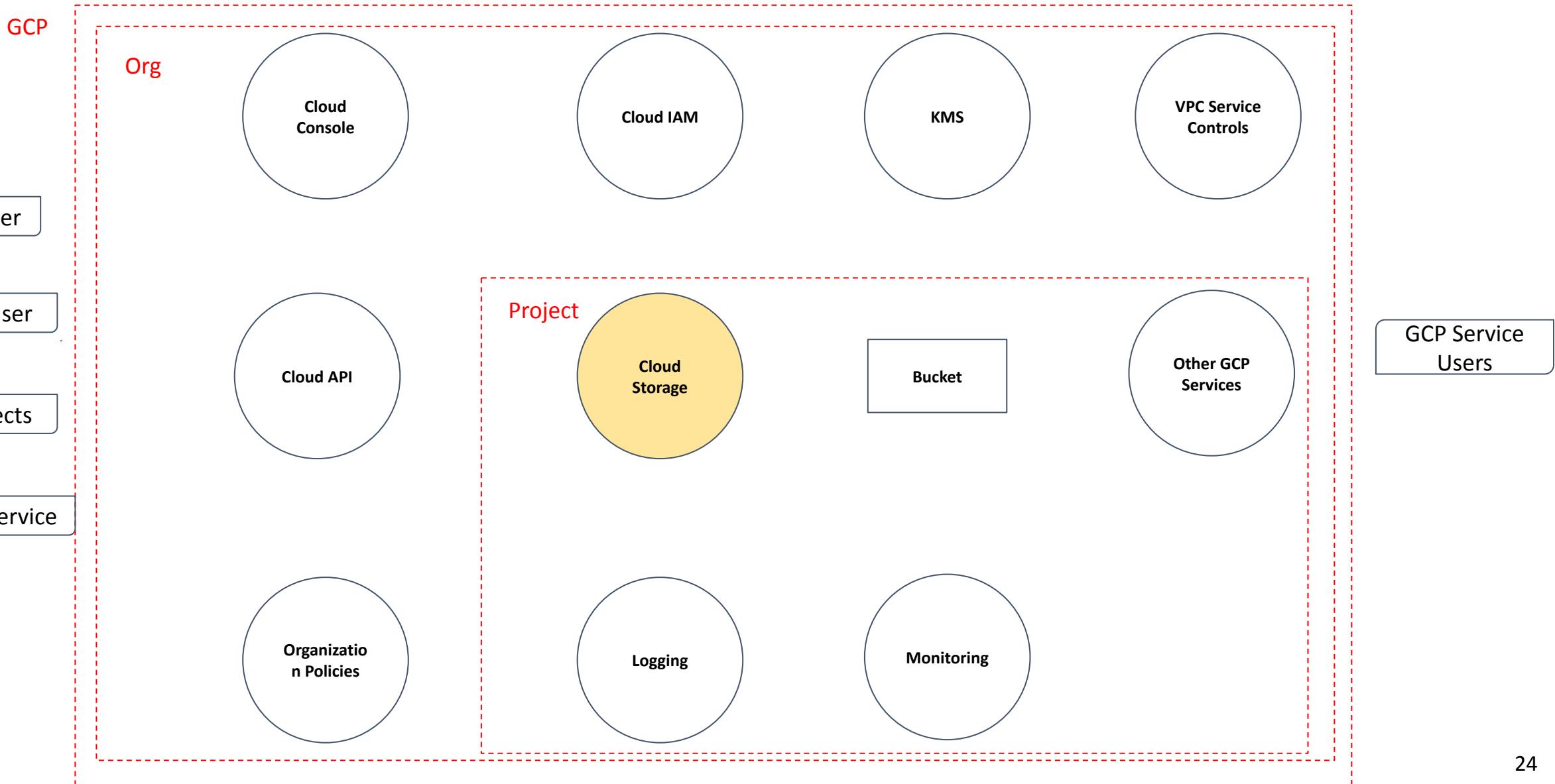


KubeCon



CloudNativeCon

North America 2023



Cloud Threat Modeling

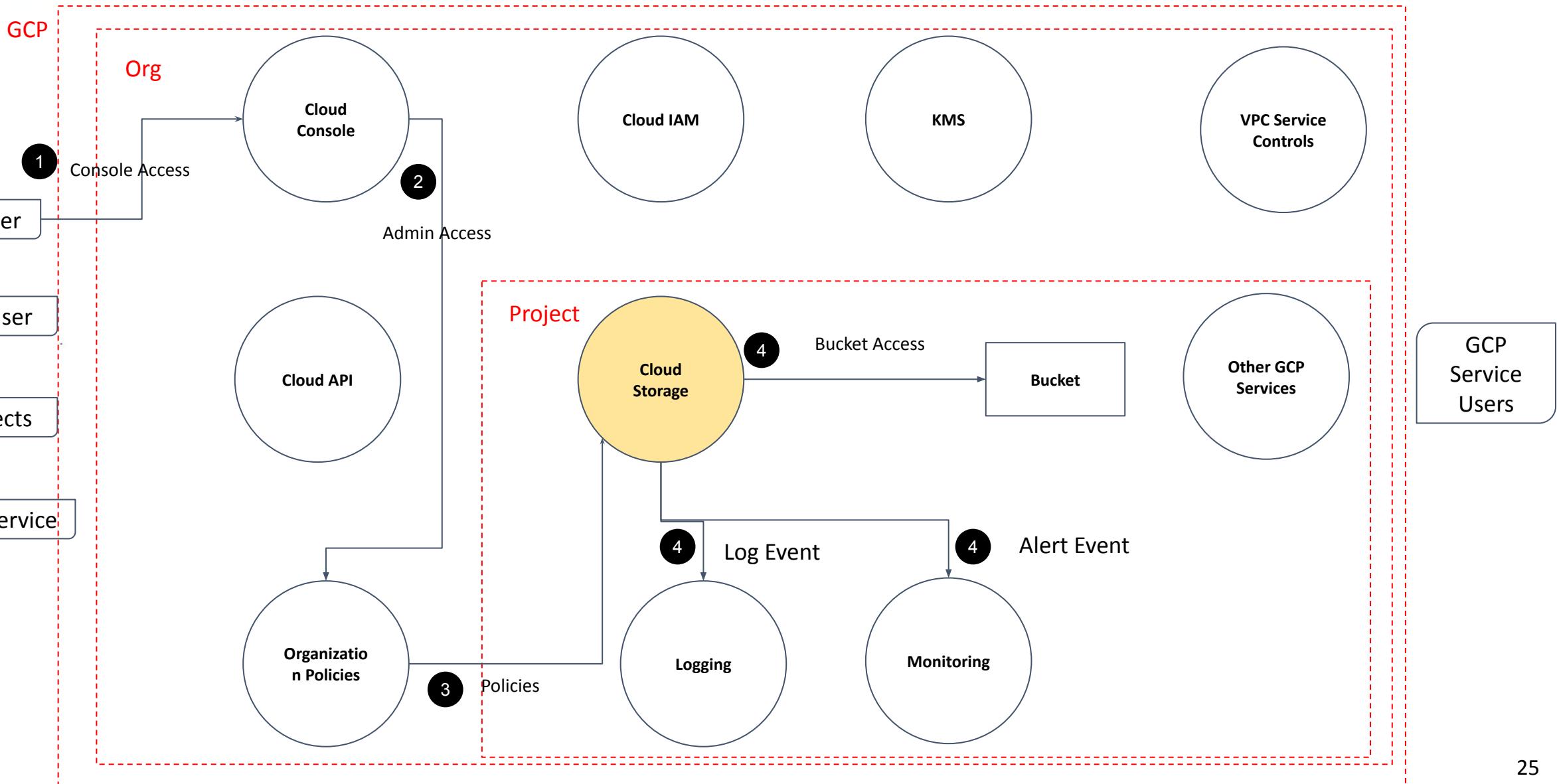


KubeCon



CloudNativeCon

North America 2023



Cloud Threat Modeling

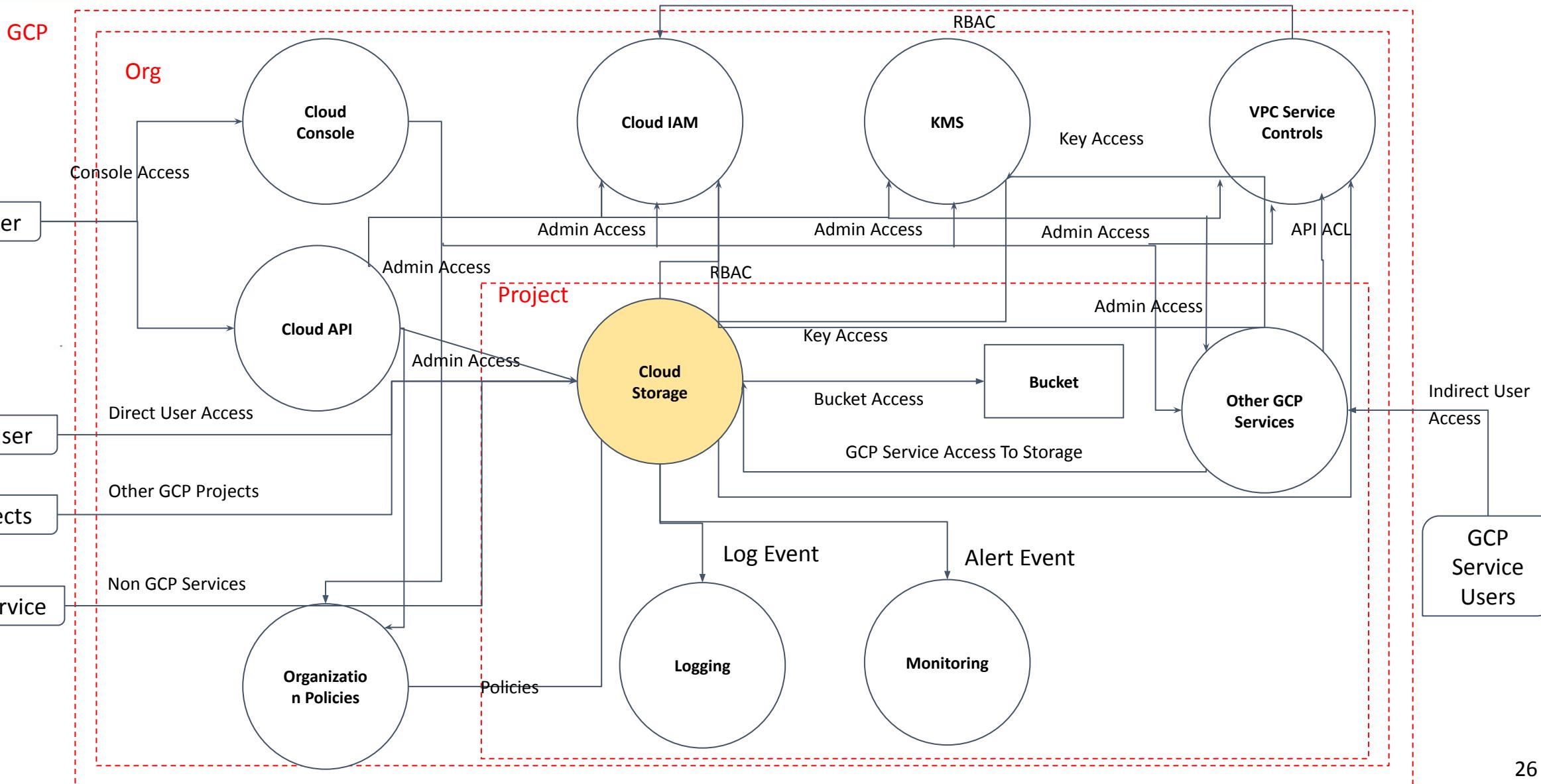


KubeCon



CloudNativeCon

North America 2023



Cloud Threat Modeling

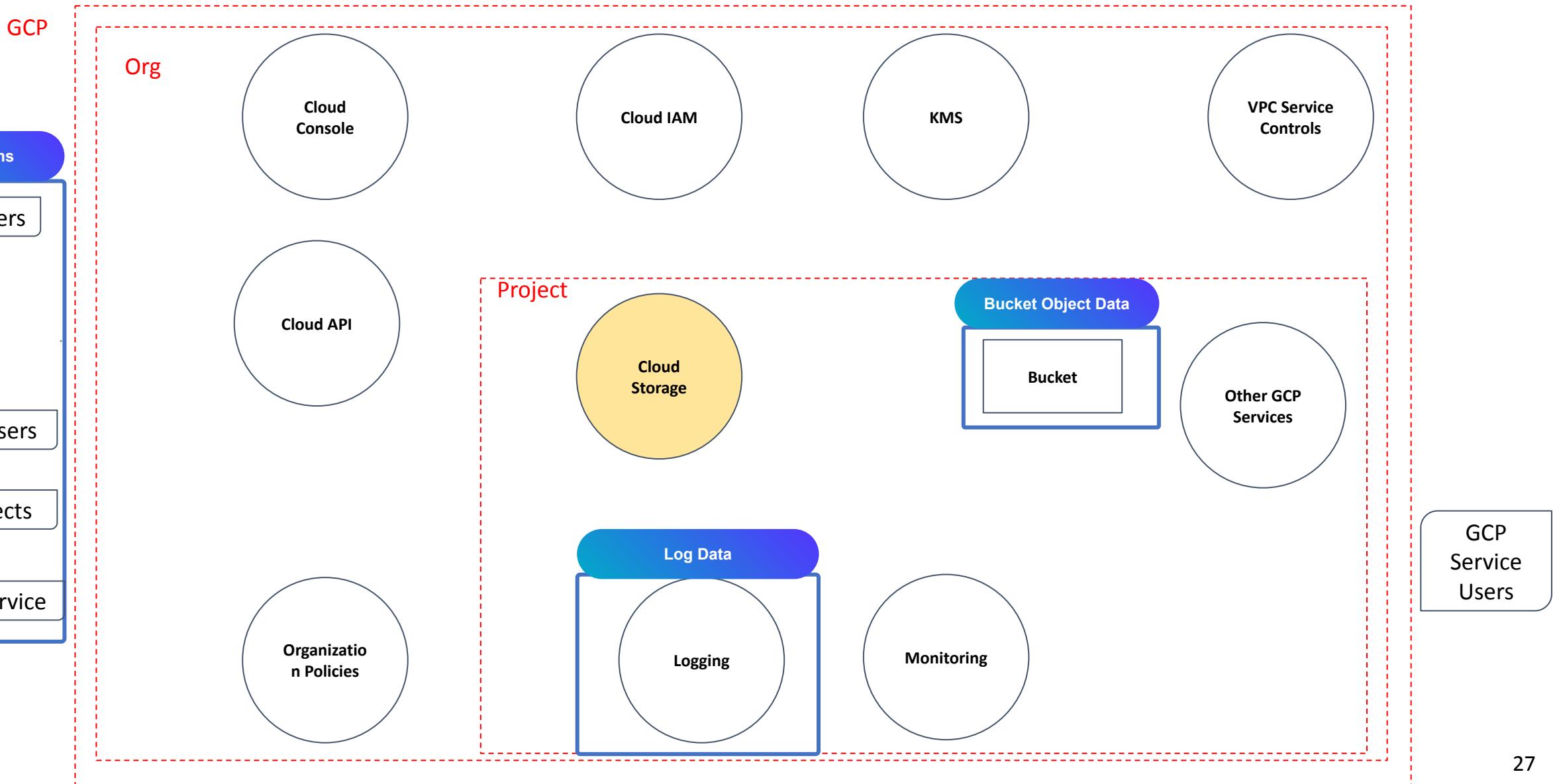


KubeCon



CloudNativeCon

North America 2023



Cloud Threat Modeling

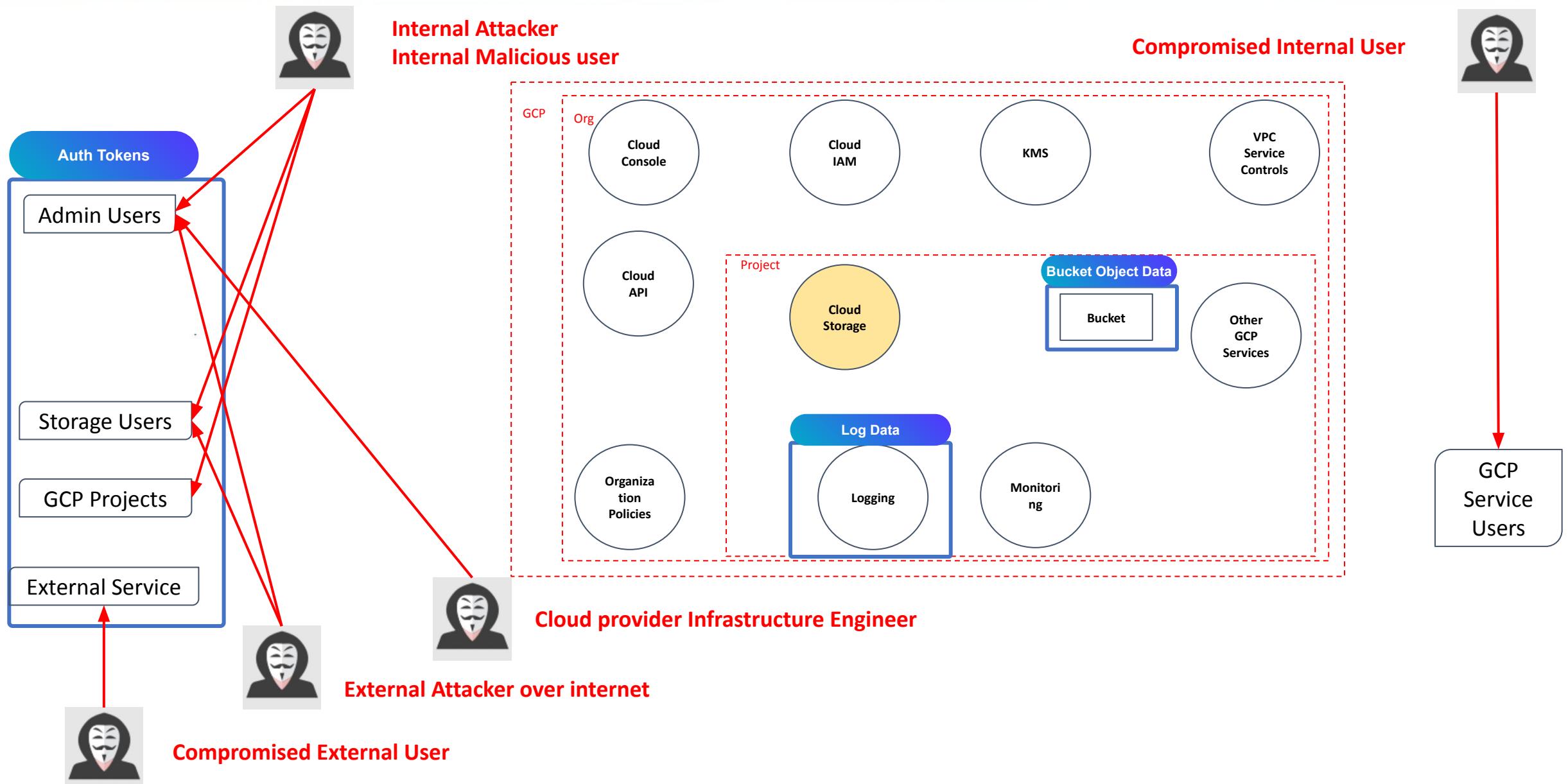


KubeCon

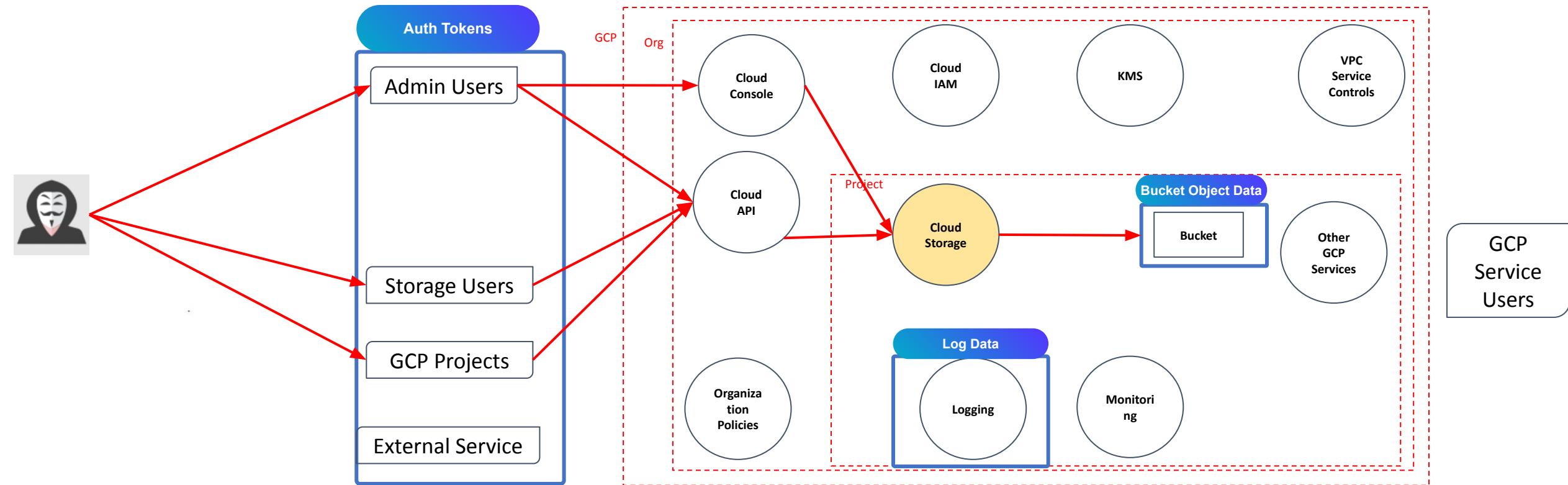


CloudNativeCon

North America 2023



Cloud Threat Modeling



Threat: Theft of credentials or access tokens

Threat Actors: Internal attacker | Internal malicious user | External attacker over the Internet

Asset: Bucket

Impact: bucket read/write permissions | Modify bucket security setting (Admin)

STRIDE Category: Spoofing, EoP

Threat Mitigation

Threat	Theft of credentials or access tokens
Threat Actors	<ul style="list-style-type: none">Internal attackerInternal malicious userExternal attacker over the Internet
Asset	Bucket
Impact	<ul style="list-style-type: none">Bucket read/write permissionsModify bucket security setting (Admin)
STRIDE Category	Spoofing, EoP



- Enable MFA
- Strong password policy
- Ensure roles are granted to principals than using primitive roles.
- Restrict VPC Service Controls with trusted IP addresses.
- Configure Google Security Command Center for cloud storage.
- Configure logs and enable alerting.

Cloud Native Adversary

Mitre ATT&CK framework

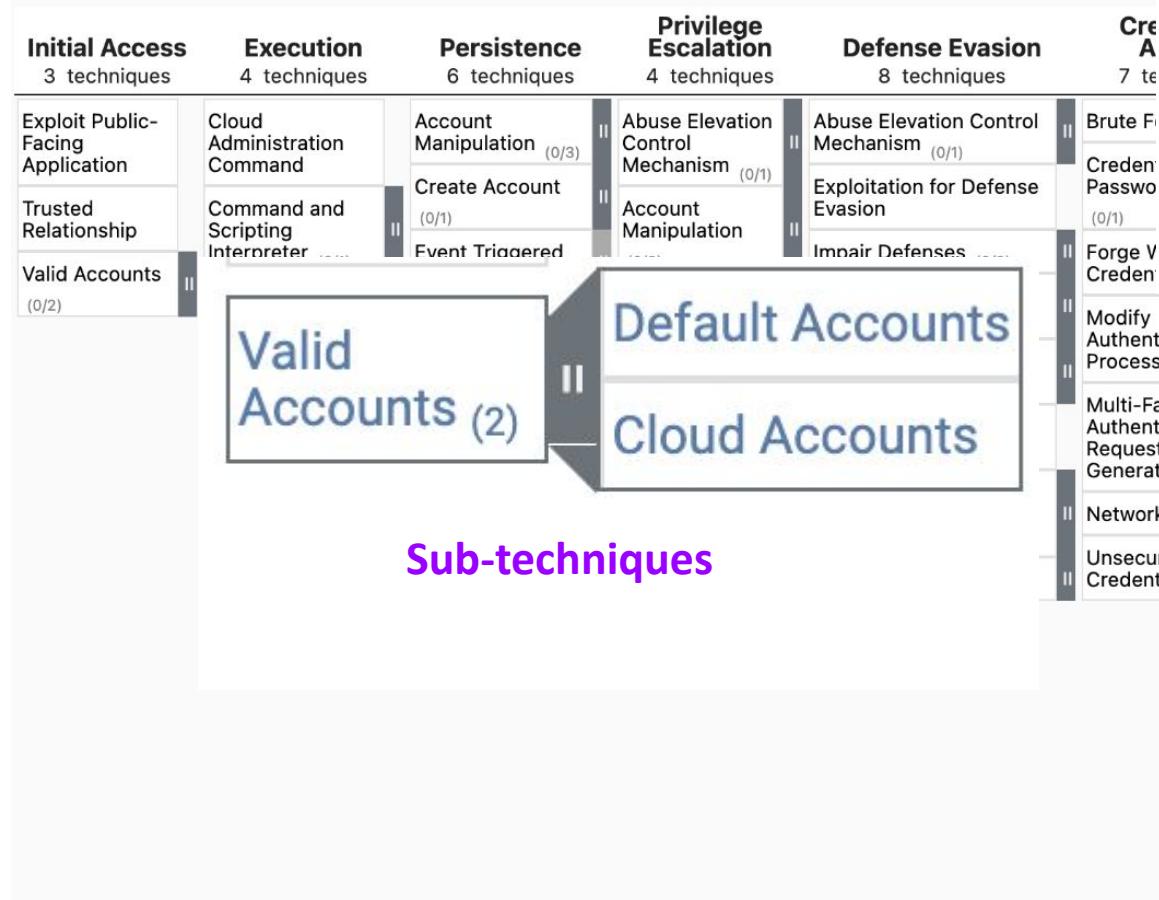
- Mitre ATT&CK framework: The framework provides a common language and understanding of adversary behavior, which can help organizations defend against potential cyber attacks, and improve their overall security posture.
- ATT&CK is maintained by MITRE, a non-profit organization that operates research and development centers for the U.S. government.

MITRE | ATT&CK®

Mitre ATT&CK framework

Tactics {

Techniques



Procedure Examples

ID	Name	Description
G1016	FIN13	FIN13 has leveraged default credentials for authenticating myWebMethods (WMS) and QLogic web management interface to gain initial access. ^[5]
S0537	HyperStack	HyperStack can use default credentials to connect to IPC\$ shares on remote machines. ^[6]
G0059	Magic Hound	Magic Hound enabled and used the default system managed account, DefaultAccount, via "powershell.exe" /c net user DefaultAccount /active:yes to connect to a targeted Exchange server over RDP. ^[7]
S0603	Stuxnet	Stuxnet infected WinCC machines via a hardcoded database server password. ^[8]

Mitigations

ID	Mitigation	Description
M1027	Password Policies	Applications and appliances that utilize default username and password should be changed immediately after the installation, and before deployment to a production environment. ^[9]

Detection

ID	Data Source	Data Component	Detects
DS0028	Logon Session	Logon Session Creation	Monitor for newly constructed logon behavior across default accounts that have been activated or logged into. These audits should also include checks on any appliances and applications for default credentials or SSH keys, and if any are discovered, they should be updated immediately.
DS0002	User Account	User Account Authentication	Monitor for an attempt by a user to gain access to a network or computing resource, often by providing credentials



Containers Adversary

Mitre Containers Matrix: <https://attack.mitre.org/matrices/enterprise/containers/>

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Impact
3 techniques	4 techniques	6 techniques	5 techniques	7 techniques	3 techniques	3 techniques	1 techniques	5 techniques
Exploit Public-Facing Application	Container Administration Command	Account Manipulation (1)	Account Manipulation (1)	Build Image on Host	Brute Force (3)	Container and Resource Discovery	Use Alternate Authentication Material (1)	Data Destruction
External Remote Services	Deploy Container	Create Account (1)	Escape to Host	Deploy Container	Steal Application Access Token	Network Service Discovery		Endpoint Denial of Service
Valid Accounts (2)	Scheduled Task/Job (1)	External Remote Services	Exploitation for Privilege Escalation	Impair Defenses (1)	Unsecured Credentials (2)	Permission Groups Discovery		Inhibit System Recovery
	User Execution (1)	Implant Internal Image	Scheduled Task/Job (1)	Indicator Removal				Network Denial of Service
		Scheduled Task/Job (1)	Valid Accounts (2)	Masquerading (1)				Resource Hijacking
		Valid Accounts (2)		Use Alternate Authentication Material (1)				
				Valid Accounts (2)				

Kubernetes Adversary



KubeCon



CloudNativeCon

North America 2023

<https://microsoft.github.io/Threat-Matrix-for-Kubernetes/>

Microsoft Threat Matrix for Kubernetes

Tactics Mitigations About Search

Tactics

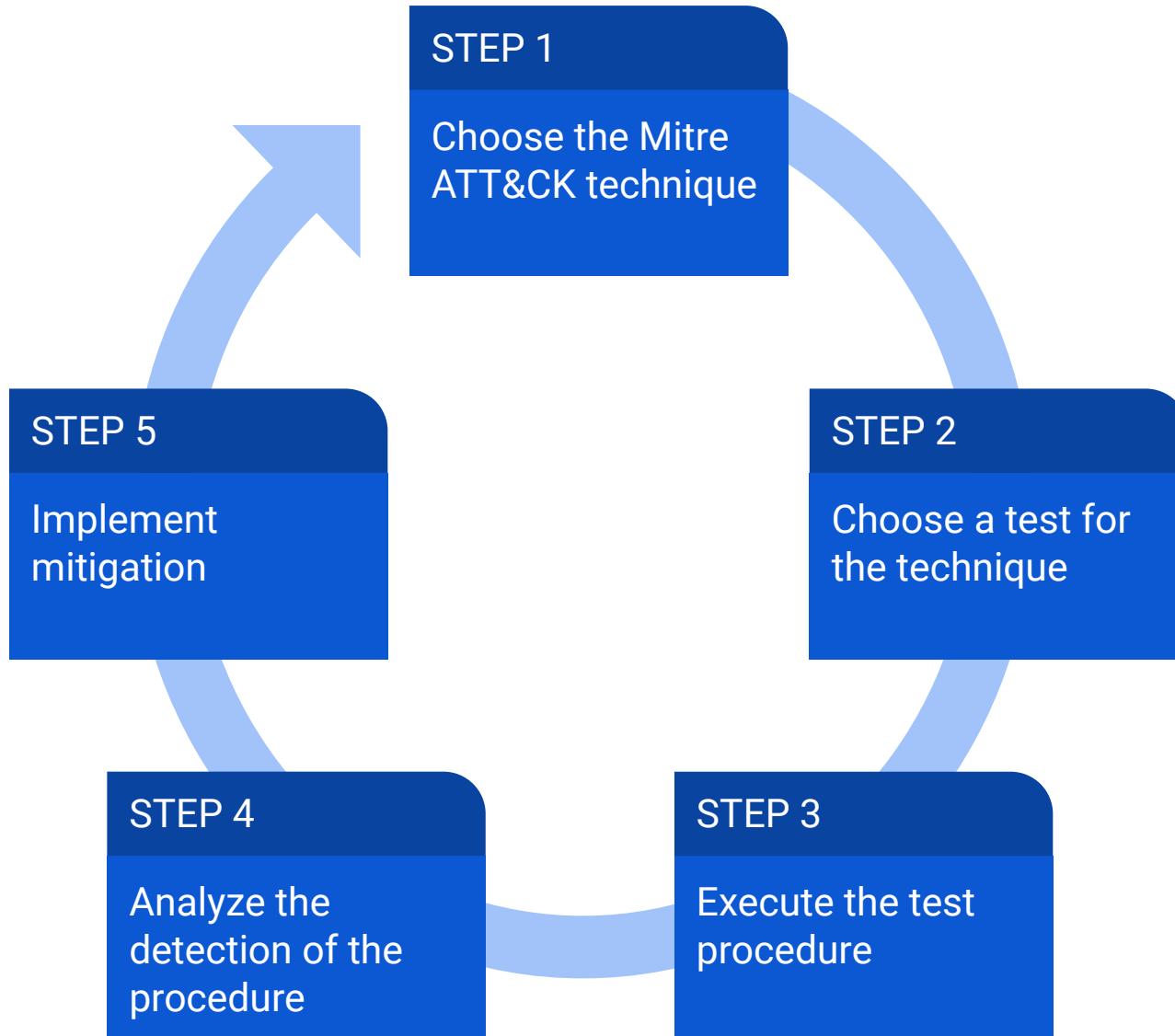
	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Impact
Initial Access	Using cloud credentials	Exec into container	Backdoor container	Privileged container	Clear container logs	List K8S secrets	Access Kubernetes API server	Access cloud resources	Images from a private registry	Data destruction
Execution	Compromised image	bash/cmd inside container	Writable hostPath mount	Cluster-admin binding	Delete K8S events	Mount service principal	Access Kubelet API	Container service account	Collecting data from pod	Resource hijacking
Persistence	In registry	New container	Kubernetes CronJob	hostPath mount	Pod / container name similarity	Container service account	Network mapping	Cluster internal networking		Denial of service
Privilege Escalation	Kubeconfig file	Application exploit (RCE)	Malicious admission controller	Access cloud resources	Connect from proxy server	Application credentials in configuration files	Exposed sensitive interfaces	Application credentials in configuration files		
Defense Evasion	Exposed sensitive interfaces	SSH server running inside container	Container service account			Access managed identity credentials	Instance Metadata API	Writable hostPath mount		
Credential Access		Sidecar injection	Static pods		Malicious admission controller		CoreDNS poisoning			
Discovery							ARP poisoning and IP spoofing			
Lateral Movement										
Collection										
Impact										

Cloud Adversary

ATT&CK Cloud <https://attack.mitre.org/matrices/enterprise/cloud/>

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Impact
5 techniques	4 techniques	7 techniques	5 techniques	12 techniques	11 techniques	14 techniques	4 techniques	5 techniques	3 techniques	9 techniques
Drive-by Compromise	Cloud Administration Command	Account Manipulation (0/5)	Abuse Elevation Control Mechanism (0/1)	Abuse Elevation Control Mechanism (0/1)	Brute Force (0/4)	Account Discovery (0/2)	Internal Spearphishing	Automated Collection	Exfiltration Over Alternative Protocol (0/0)	Account Access Removal
Exploit Public-Facing Application	Create Account (0/1)	Account Manipulation (0/5)	Domain Policy Modification (0/1)	Credentials from Password Stores (0/1)	Cloud Infrastructure Discovery	Remote Services (0/2)	Data from Cloud Storage	Taint Shared Content	Exfiltration Over Web Service (0/1)	Data Destruction
Phishing (0/2)	Command and Scripting Interpreter (0/1)	Event Triggered Execution (0/0)	Exploitation for Defense Evasion	Exploitation for Credential Access	Cloud Service Dashboard	Cloud Service Discovery	Data from Information Repositories (0/3)	Data Staged (0/1)	Transfer Data to Cloud Account	Data Encrypted for Impact
Trusted Relationship	Serverless Execution	Implant Internal Image	Domain Policy Modification (0/1)	Forge Web Credentials (0/2)	Cloud Storage Object Discovery	Log Enumeration	Email Collection (0/2)			Defacement (0/1)
Valid Accounts (0/2)	User Execution (0/1)	Modify Authentication Process (0/2)	Hide Artifacts (0/1)	Impair Defenses (0/3)	Multi-Factor Authentication Request Generation	Network Service Discovery				Endpoint Denial of Service (0/3)
		Office Application Startup (0/6)	Event Triggered Execution (0/0)	Impersonation	Network Sniffing	Network Sniffing				Financial Theft
		Valid Accounts (0/2)	Valid Accounts (0/2)	Indicator Removal (0/1)	Steal Application Access Token	Password Policy Discovery				Inhibit System Recovery
				Modify Authentication Process (0/2)	Steal or Forge Authentication Certificates	Permission Groups Discovery (0/1)				Network Denial of Service (0/2)
				Modify Cloud Compute Infrastructure (0/5)	Steal Web Session Cookie	Software Discovery (0/1)				Resource Hijacking
				Unused/Unsupported Cloud Regions	Unsecured Credentials (0/3)	System Information Discovery				
				Use Alternate Authentication Material (0/2)						
				Valid Accounts (0/2)						

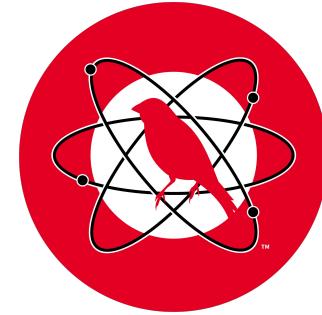
Cloud Attack Emulation Workflow



Tools

Tooling: Atomic Red Team

- Atomic Red Team™ (<https://github.com/redcanaryco/atomic-red-team>):
 - An open source framework
 - A library of tests mapped to the [MITRE ATT&CK®](#) framework



Coverage	Cloud Infrastructure AWS, Azure and GCP	Kubernetes	Containers
----------	--	------------	------------

Atomic Red Team - Example

T1098.001: Additional Cloud Credentials

CREATE THE KEY

```
aws iam create-access-key --user-name #{username} >  
$PathToAtomicsFolder/T1098.001/bin/aws_secret.creds  
cd $PathToAtomicsFolder/T1098.001/bin/  
../aws_secret.sh
```

SAVE THE KEY

DISPLAY THE KEY

Kubernetes Threat Matrix



KubeCon



CloudNativeCon

North America 2023

<https://kubernetes-threat-matrix.redguard.ch/persistence/backdoor-container/>

Initial access	Execution	Persistence	Privilege escalation	Defense evasion	Credential access	Discovery	Lateral movement	Collection	Impact
			Backdoor Container						
Using Cloud	Exec into Container	Backdoor Container							
			Attackers run their malicious code in a container in the cluster. By using the Kubernetes controllers such as DaemonSets or Deployments, attackers can ensure that number of containers run in one, or all, the nodes in the cluster.						
Compromised images in registry	bash/cmd in container	Writable hostPath mount			Example				
			One of the simplest and on the same time most efficient ways to deploy backdoor containers is the usage of a <i>DaemonSet</i> . The following adds the attacker's SSH key to the authorized_keys file on every node in the cluster every 10 minutes (600 seconds). As a <i>DaemonSet</i> automatically makes sure to deploy a pod on each node, this means new added nodes will immediately be compromised when they become available.						
Kubeconfig file	New container	Kubernetes CronJob			<pre>apiVersion: apps/v1 kind: DaemonSet metadata: name: evil-daemonset labels: app: evil-daemonset spec: selector: matchLabels: app: evil-daemonset template: metadata: labels: app: evil-daemonset spec: containers: - name: evil image: ubuntu command: ["/bin/sh", "-c", "mkdir -p /host/root/.ssh && echo 'ssh-rsa AAAAB3NzaC1y...CUkwfwh+iSTP' >> /host/root/.ssh/authorized_keys && sleep 600"]</pre>				
Application vulnerability	Application exploit (RCE)	Malicious admission controller							
Exposed sensitive interfaces	SSH server running in inside container								
		Sidecar injection							

Please note that this is a quick and dirty example only made for demonstration purposes and the `authorized_keys` file will be filled up over time.

IP SPOOFING

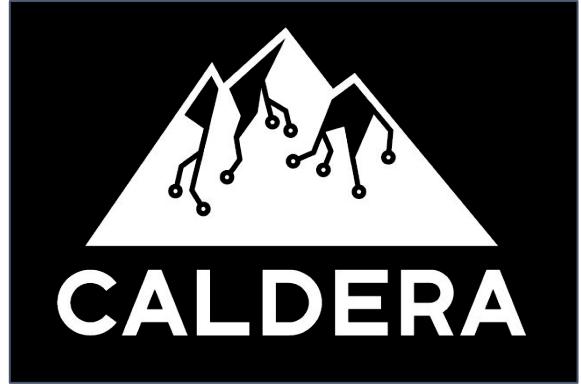
Tooling: CALDERA

- MITRE Caldera™ (<https://github.com/mitre/caldera>) is an automated adversary emulation tool:
 - Built-in behaviors mapped to ATT&CK techniques
 - Automate sequences of behaviors

The screenshot shows the Caldera web interface with a dark theme. On the left is a sidebar with navigation links: agents, abilities, adversaries, operations (which is selected), atomic, campaigns, plugins (access, atomic, compass, debrief, fieldmanual, manx, sandcat, stockpile, training), configuration (fact sources, objectives, planners, contacts, obfuscators, configuration, exfilled files, api docs), and a log out button. The main area is titled 'Operations' and shows a 'Worm Operation'. It displays a timeline of events:

Decide	Status	Link/Ability Name	Agent #/paw	Host	pid	Link Command	Link Output
3/12/2022, 11:07:18 AM GMT -7	running	Collect ARP details	nxdgkk	VAN-DST-10	8232	View Command	View Output
3/12/2022, 11:07:39 AM GMT -7	running	Reverse nslookup IP	nxdgkk	VAN-DST-10	6188	View Command	View Output
3/12/2022, 11:08:19 AM GMT -7	collect	Reverse nslookup IP	nxdgkk	VAN-DST-10	n/a	View Command	No output.

At the top of the main area, there are buttons for 'Download', 'Delete', 'Current state: running', 'Stop', 'Pause', 'Run 1 Link', 'Obfuscation: plain-text', 'Manual', and 'Autonomous'. There are also buttons for '+ Manual Command' and '+ Potential Link'.



ATT&CK Workbench

An application allowing users to **explore, create, annotate, and share** extensions of the MITRE ATT&CK® knowledge base.

The screenshot shows the ATT&CK WORKBENCH v1.0.0 interface. The main page displays the 'Screen Capture' technique. Key details include:

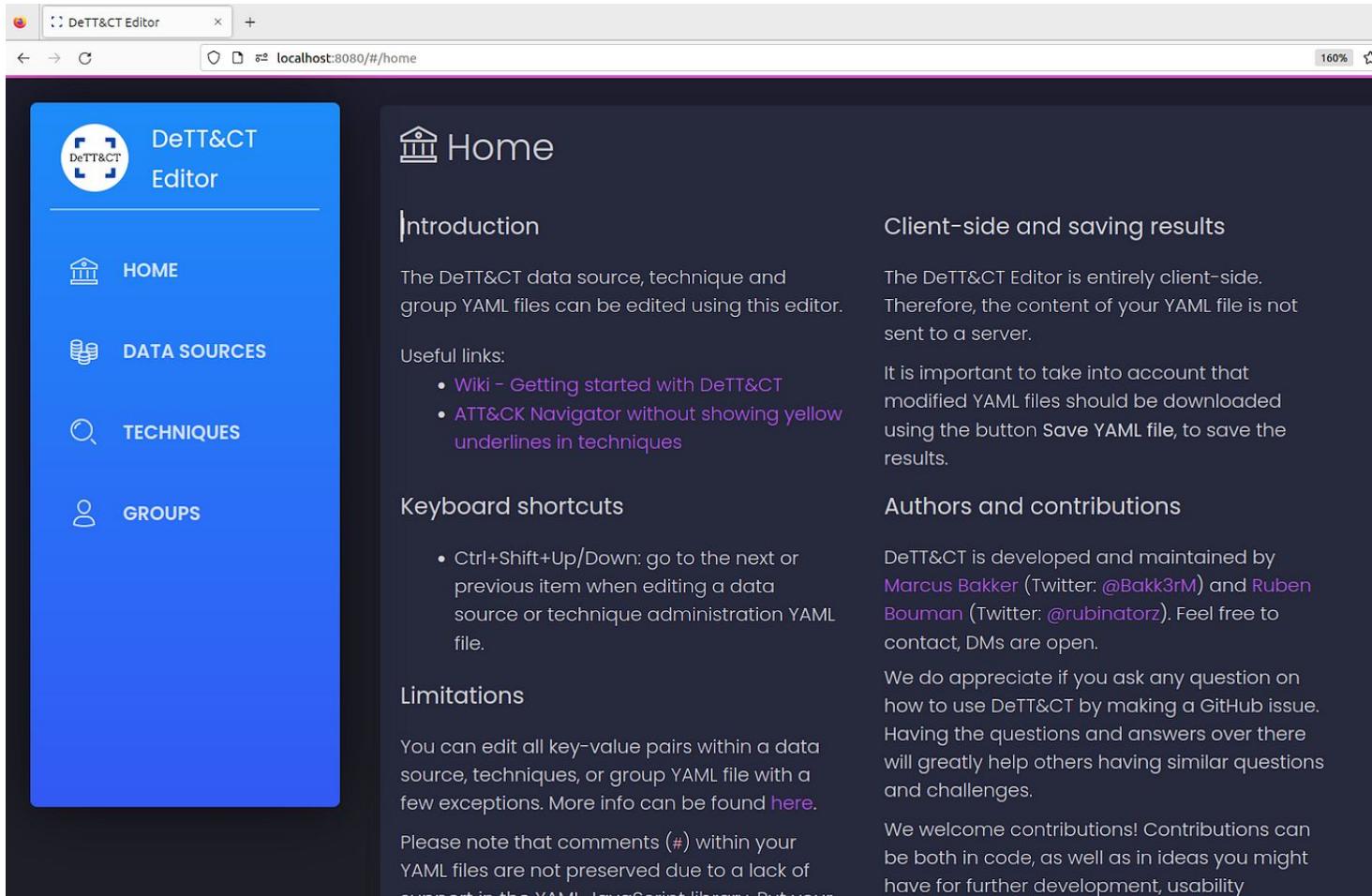
- ID:** T1113
- VERSION:** 1.1
- sub-technique?**: Unchecked
- PLATFORMS:** Linux, macOS, Windows
- CAPEC IDs:** CAPEC-648
- DESCRIPTION:** Adversaries may attempt to take screen captures of the desktop to gather information over the course of an operation. Screen capturing functionality may be included as a feature of a remote access tool used in post-compromise operations. Taking a screenshot is also typically possible through native utilities or API calls, such as `CopyFromScreen`, `xwd`, or `screencapture`.^{[1][2]}
- DOMAINS:** enterprise-attack
- DATA SOURCES:** API monitoring, Process monitoring, File monitoring
- TACTICS:** collection

On the right side, there is a **NOTES** section with a search bar and a note titled "Data Collection" from 15 JUNE 2021, 9:04 AM. The note content is: "Review data source information and create a plan to start collecting data required to detect this technique."

Detect Tactics, Techniques & Combat Threats.

It helps Blue Team using the MITRE ATT&CK framework:

- Detect gaps in detection coverage or visibility.
- Prioritize the ingestion of new log sources.



The screenshot shows a web browser window titled "DeTT&CT Editor" with the URL "localhost:8080/#/home". The left sidebar has a blue header with the "DeTT&CT Editor" logo and four menu items: HOME, DATA SOURCES, TECHNIQUES, and GROUPS. The main content area has a dark background with white text. It includes sections for "Home", "Introduction", "Useful links", "Keyboard shortcuts", "Limitations", and "Authors and contributions".

- Home**
- Introduction**

The DeTT&CT data source, technique and group YAML files can be edited using this editor.
- Useful links**
 - [Wiki - Getting started with DeTT&CT](#)
 - [ATT&CK Navigator without showing yellow underlines in techniques](#)
- Keyboard shortcuts**
 - Ctrl+Shift+Up/Down: go to the next or previous item when editing a data source or technique administration YAML file.
- Limitations**

You can edit all key-value pairs within a data source, techniques, or group YAML file with a few exceptions. More info can be found [here](#).
- Authors and contributions**

DeTT&CT is developed and maintained by [Marcus Bakker](#) (Twitter: [@Bakk3rM](#)) and [Ruben Bouman](#) (Twitter: [@rubinatorz](#)). Feel free to contact, DMs are open.

We do appreciate if you ask any question on how to use DeTT&CT by making a GitHub issue. Having the questions and answers over there will greatly help others having similar questions and challenges.

We welcome contributions! Contributions can be both in code, as well as in ideas you might have for further development, usability

Tooling: AWS Threat Composer



KubeCon

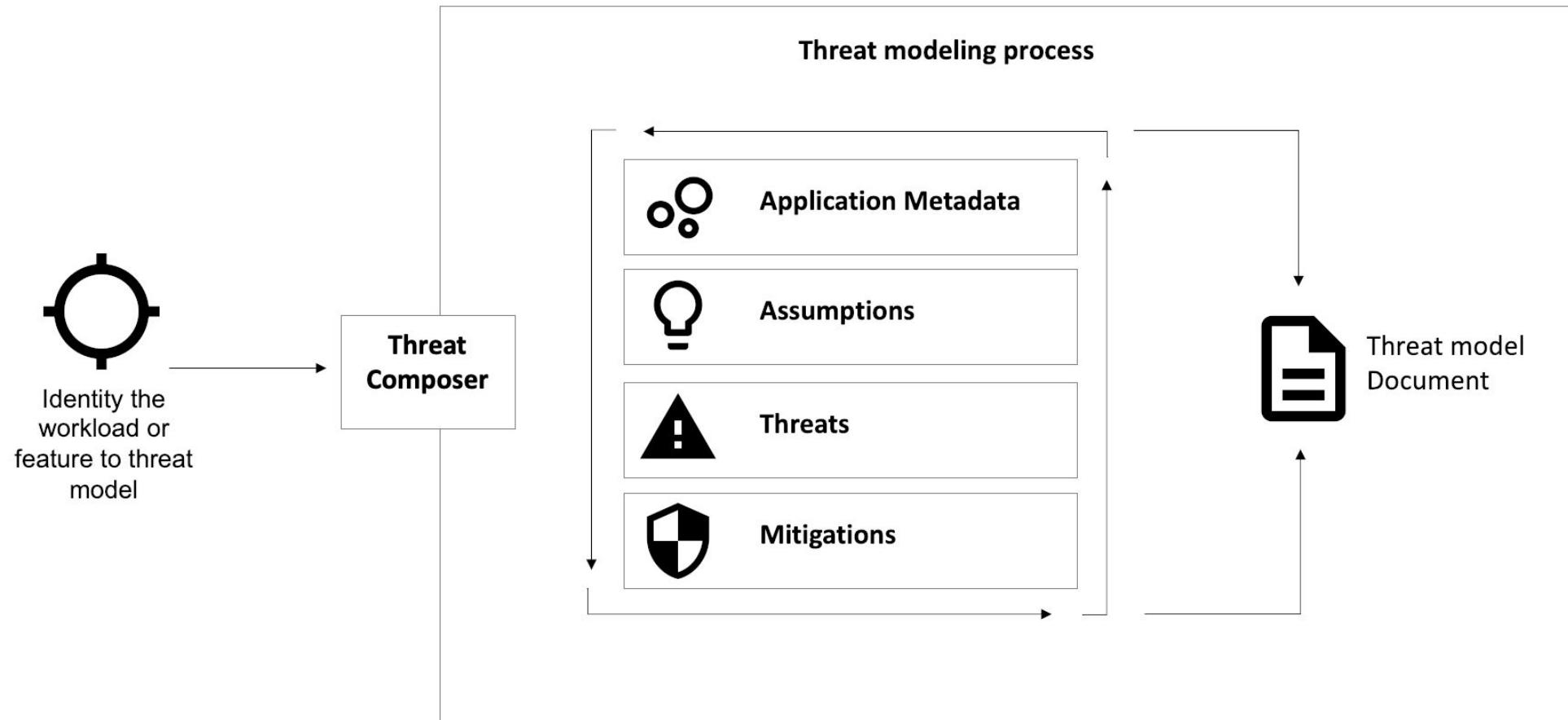


CloudNativeCon

North America 2023

<https://github.com/awslabs/threat-composer>

<https://awslabs.github.io/threat-composer/workspaces/default/dashboard?mode=Full>

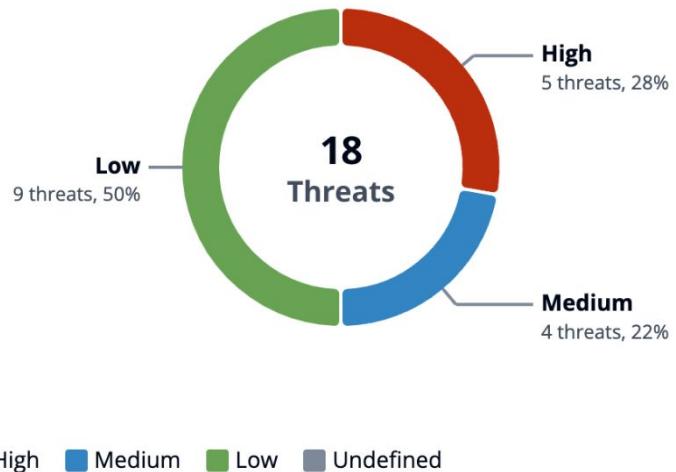


Tooling: AWS Threat Composer

Threat summary

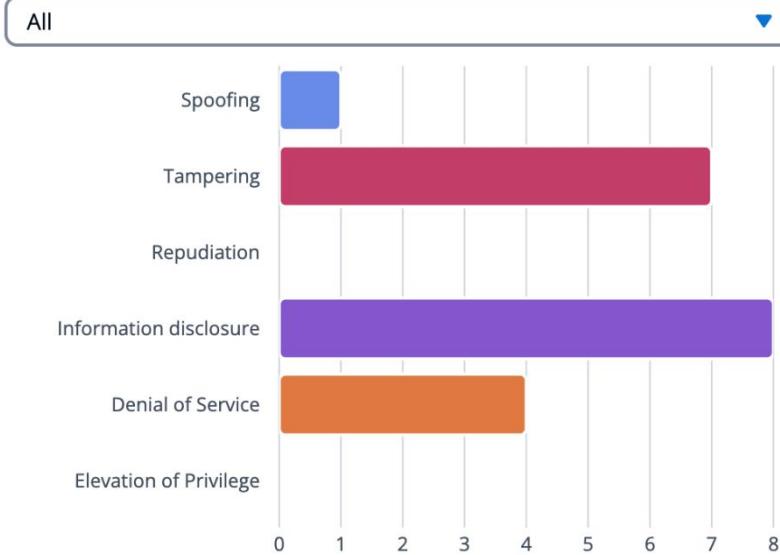
Total	No mitigation and assumption	No mitigation	High	Med	Low	Missing priority
18	0	4 ⚠	5	4	9	0

Threat prioritization



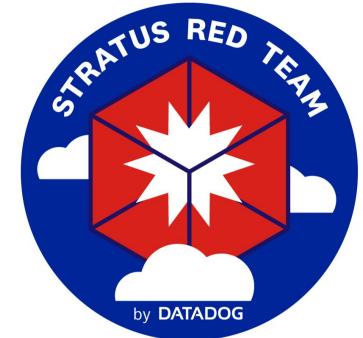
Threat category distribution

Filter by threat priority

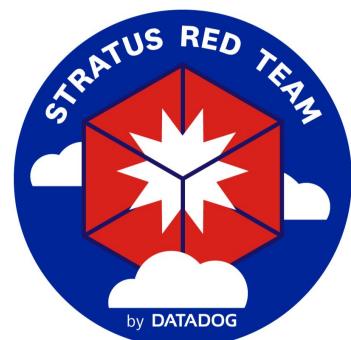


Stratus Red Team

- Stratus Red Team™: A Granular, Actionable Adversary Emulation for the Cloud
 - <https://github.com/DataDog/stratus-red-team>
 - Attack techniques mapped to MITRE ATT&CK



Coverage	AWS	GCP	Azure	Kubernetes
----------	-----	-----	-------	------------



Stratus Red Team



KubeCon



CloudNativeCon

North America 2023



Cloud Offensive Toolkits

		<p>Pacu: AWS exploitation framework, designed for offensive security testing. https://github.com/RhinoSecurityLabs/pacu</p>
		<p>Microburst: A PowerShell Toolkit for Attacking Azure. https://github.com/NetSPI/MicroBurst</p>
		<p>PowerZure: PowerShell framework to assess Azure security https://github.com/hausec/PowerZure</p>
 Google Cloud		<p>Google Cloud Platform Security Control Mappings to MITRE ATT&CK® https://center-for-threat-informed-defense.github.io/security-stack-mappings/GCP/README.html</p>

Takeaways

- Security needs to be automated in the cloud the same way you automate cloud infrastructure with Infrastructure-as-Code (Policy as Code) => **Policy Driven Security.**
- Use cloud native tools to enhance threat modeling: observability and tracing tools,
- Translating policies into consistent, effective, and actionable tasks.
- Think in graphs, not lists!

Further Reading

- [Defenders think in lists. Attackers think in graphs. As long as this is true, attackers win](#), John Lambert
- [AWS Threat Modeling workshop.](https://explore.skillbuilder.aws/learn/course/13274/play/81716/threat-modeling-for-builders)
- [Cloud Analytics Blueprint](https://github.com/center-for-threat-informed-defense/cloud-analytics)
- [Google Cloud Storage Threat Modeling](https://research.nccgroup.com/2023/01/31/threat-modelling-cloud-platform-services-by-example-google-cloud-storage/)



KubeCon



CloudNativeCon

North America 2023

Thank you! Any questions?



Please scan the QR Code above
to leave feedback on this session