Security Kill Chain Stages in a 100K+ Daily Container Environment with Falco

Natch Ruengsakulrach
Eric Hollis

Introduction

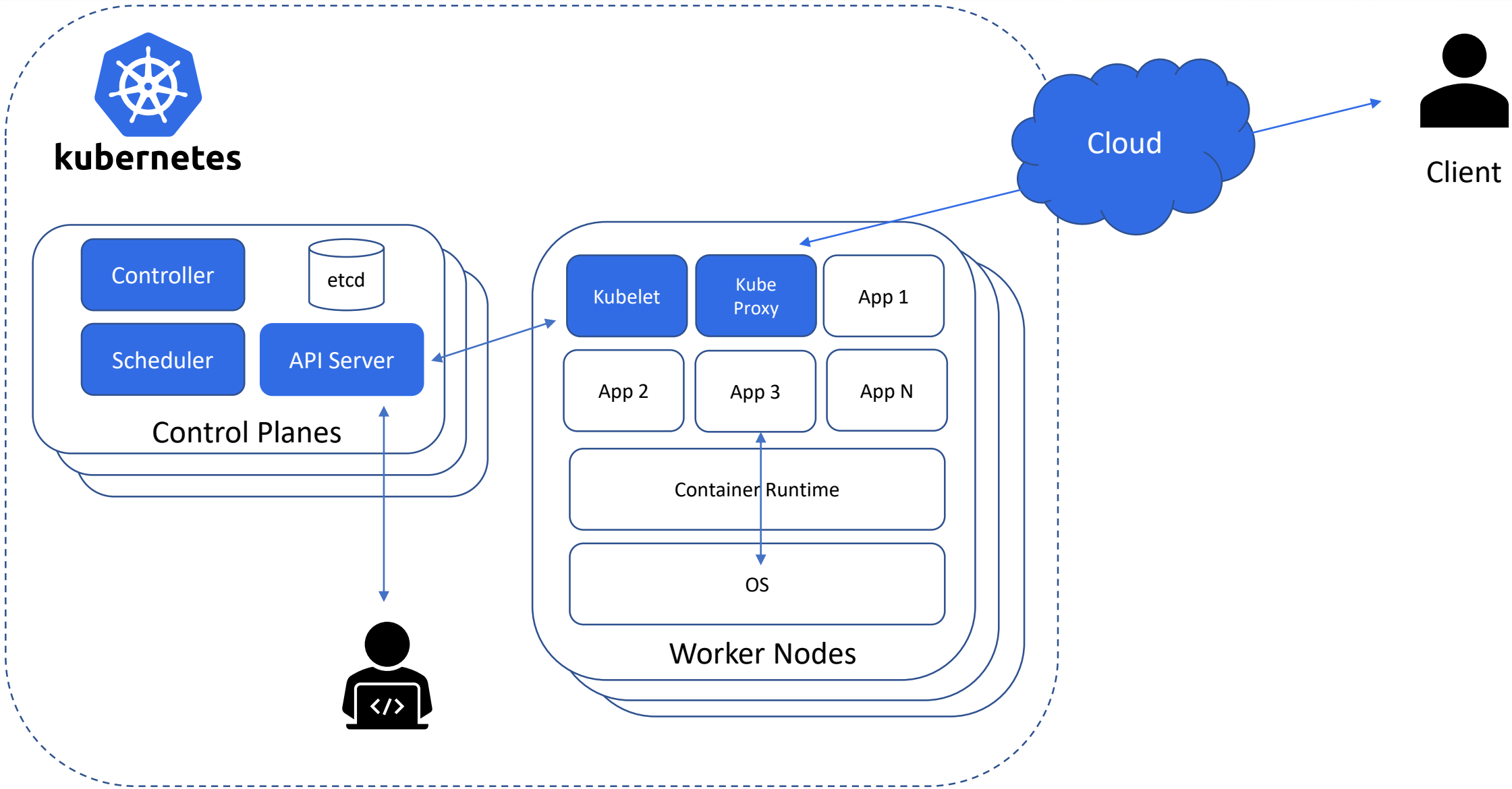
Eric Hollis, MathWorks
Twitter: @ehollis3942

- Team Lead, Extended Detection and Response
- Interested in cloud security, automation, and threat hunting

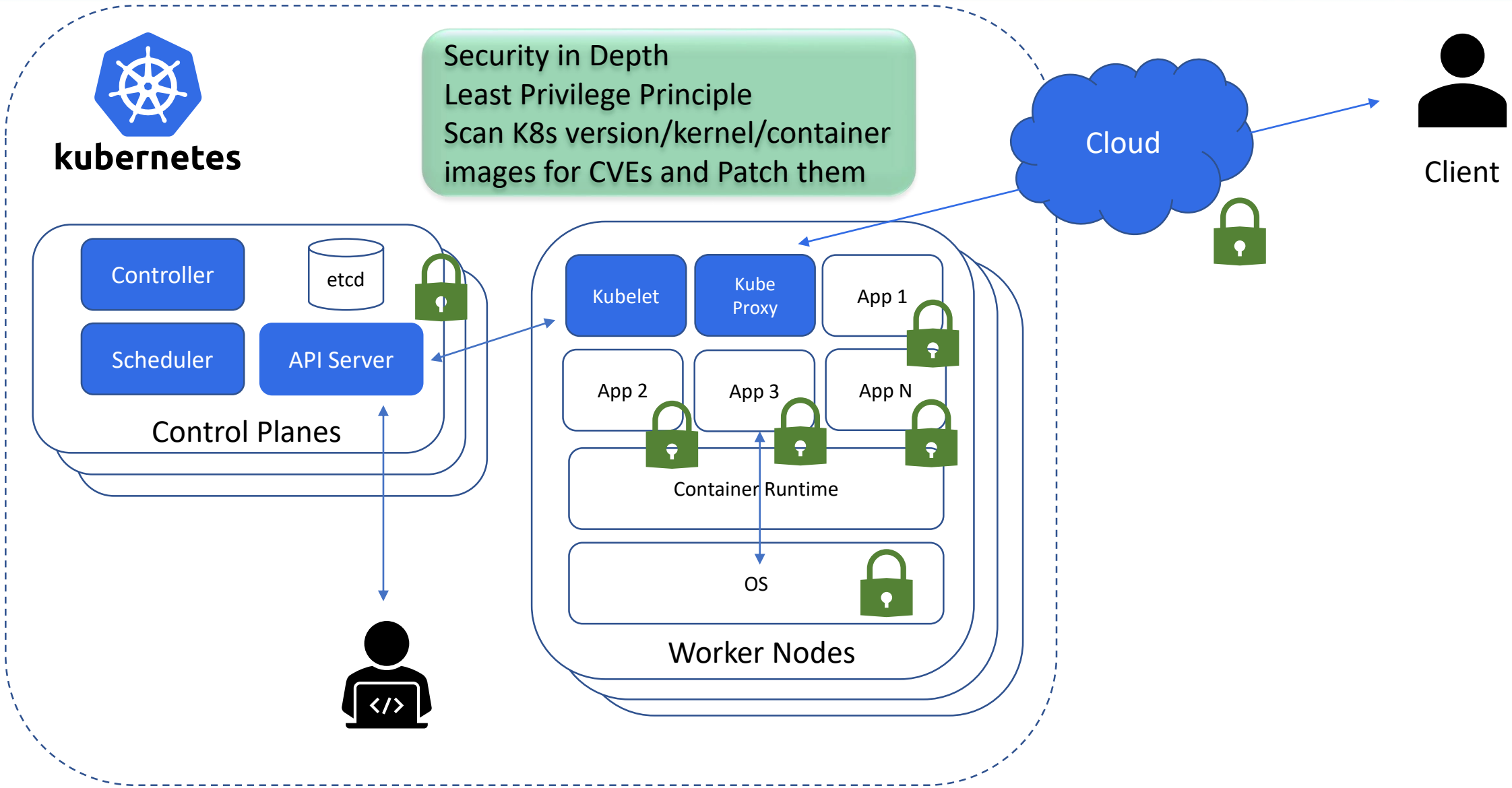
Natch Ruengsakulrach, MathWorks

- Software Engineer
- Interested in cloud-native application and distributed system design

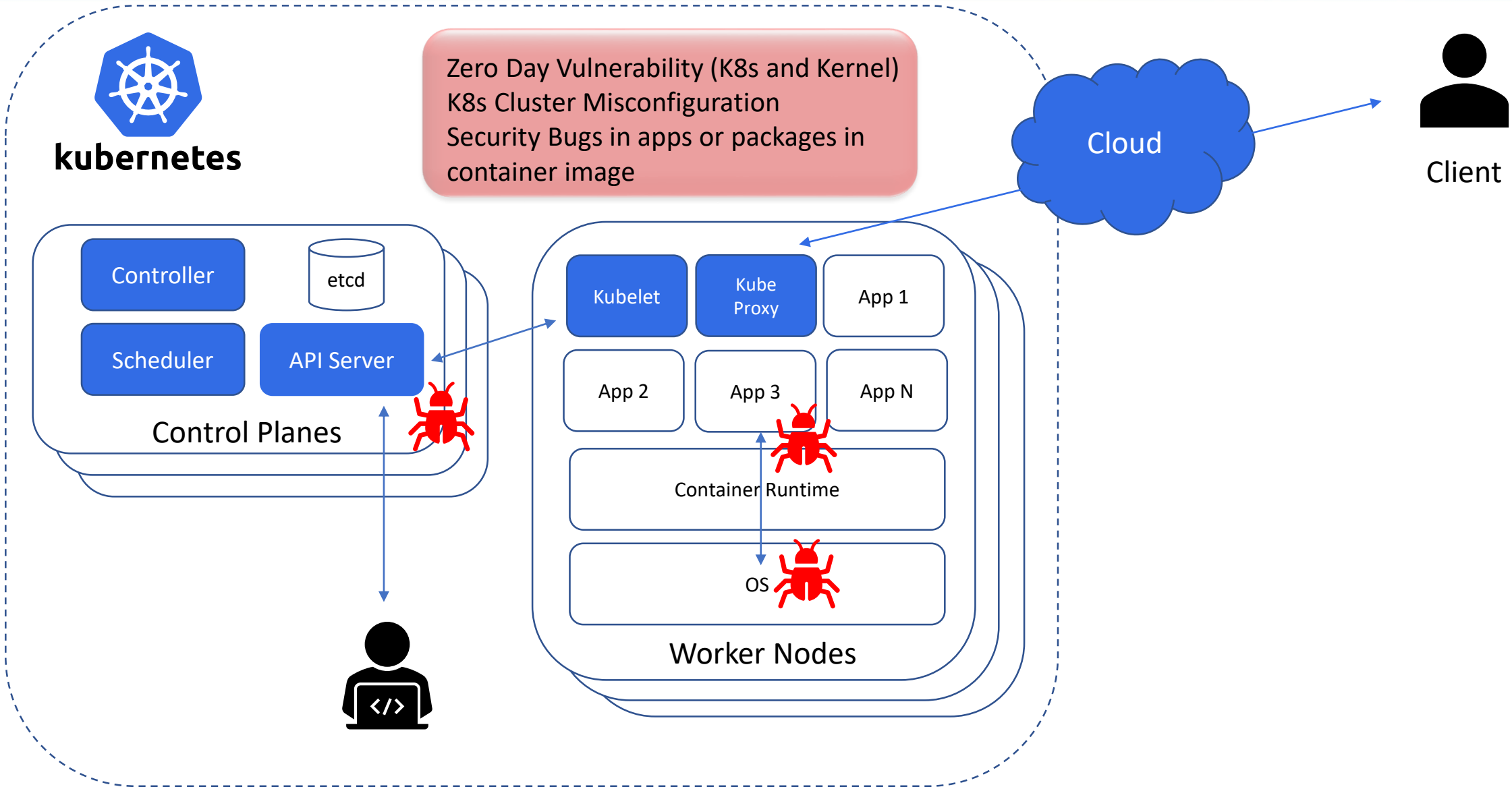
Our Cluster Architecture



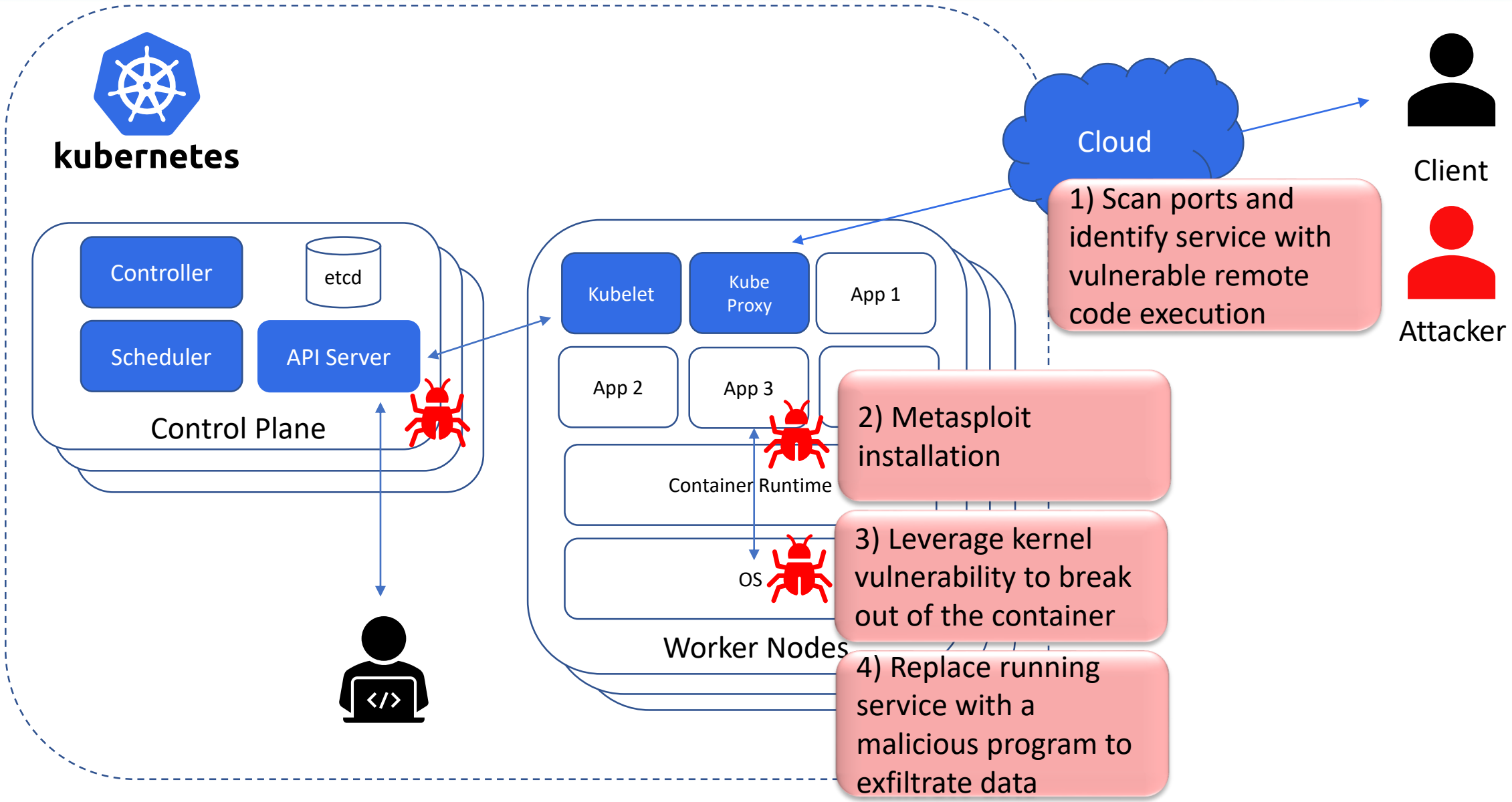
Securing Our Cluster



“No System is Perfectly Secured”



Here Comes 2020



How Do We Trace Back?

1) Scan ports and identify service with vulnerable remote code execution

What is the attackers' source IP?

2) Metasploit installation

What package, which container, and by who?

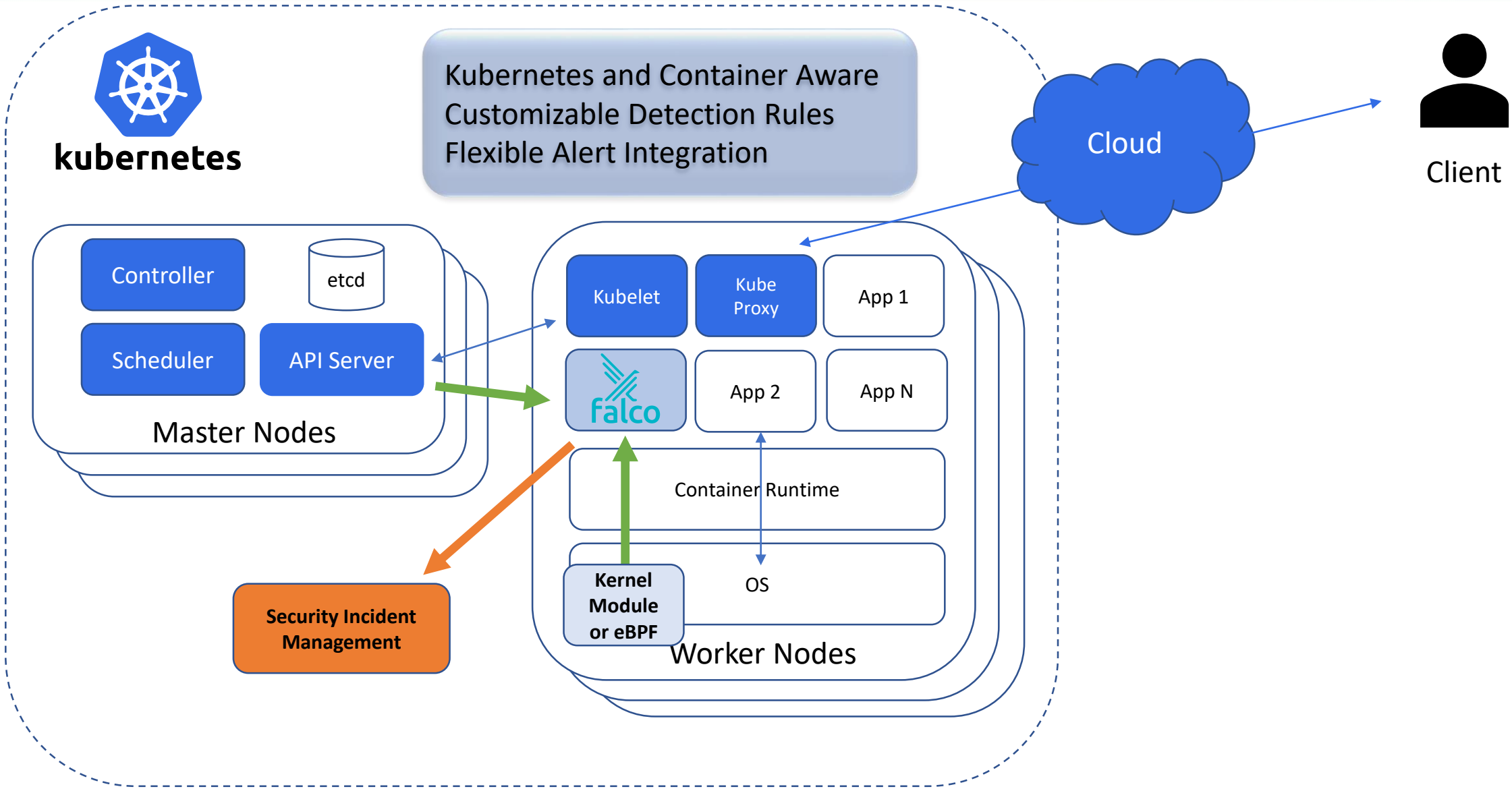
3) Leverage kernel vulnerability to break out of the container

How was the kernel vulnerability exploited?

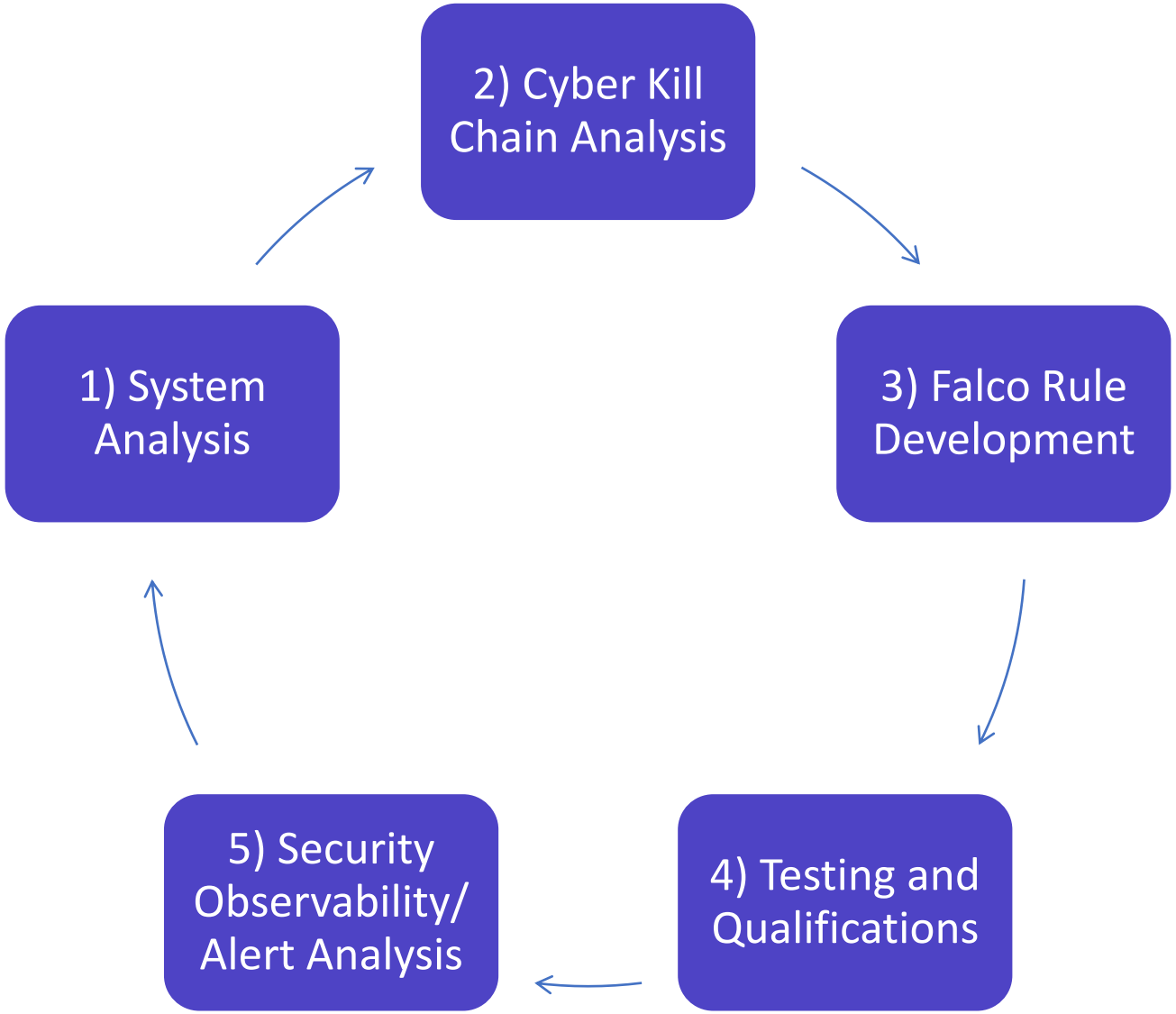
4) Replace running service with a malicious program to exfiltrate data

What service was removed? What does the malicious program do?

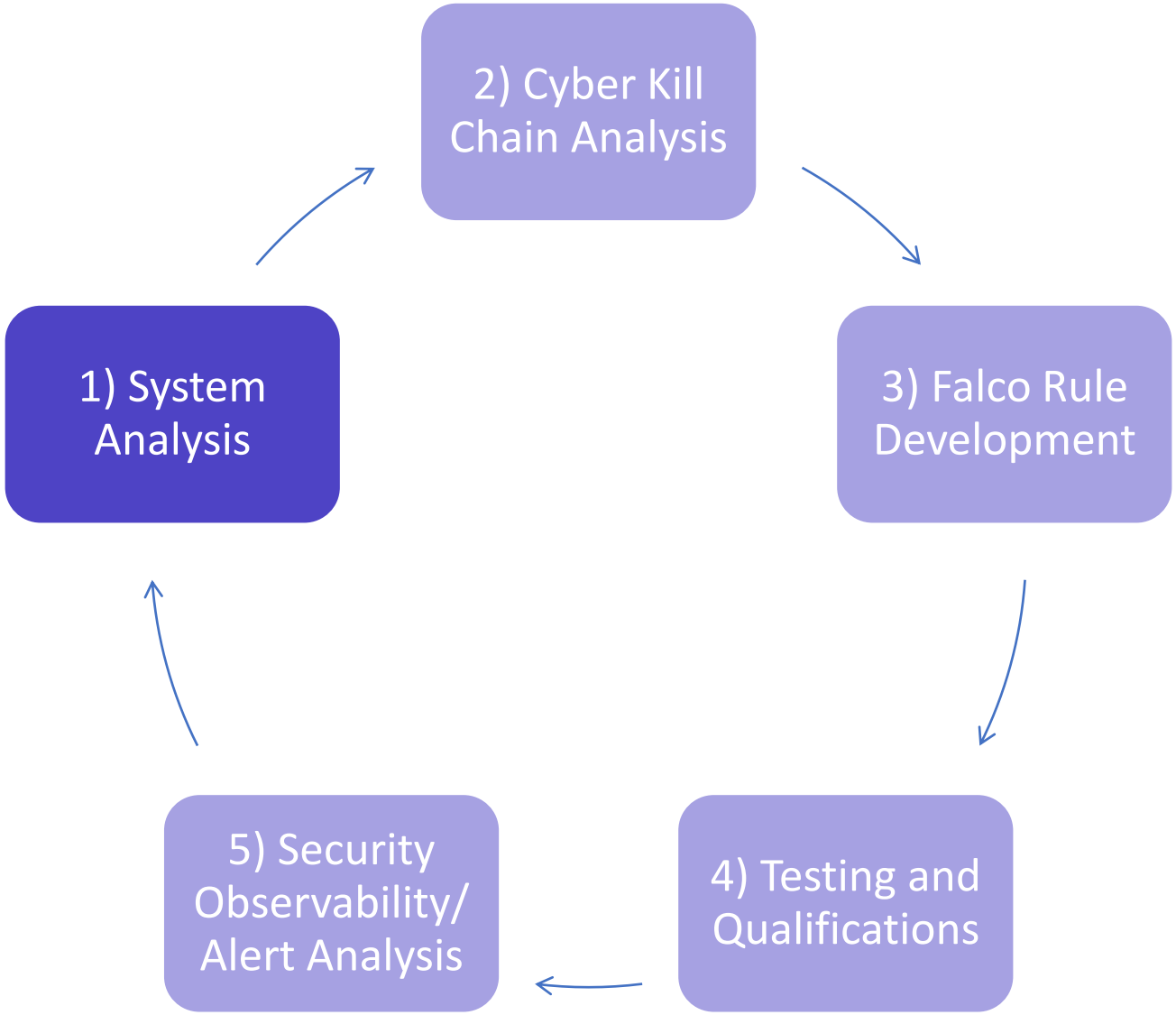
Why Falco?


kubernetes
Kubernetes and Container Aware
Customizable Detection Rules
Flexible Alert Integration
Cloud
Client
Controller
etcd
Scheduler
API Server
Master Nodes
Kubelet
Kube Proxy
App 1
falco
App 2
App N
Container Runtime
OS
Kernel Module or eBPF
Worker Nodes
Security Incident Management

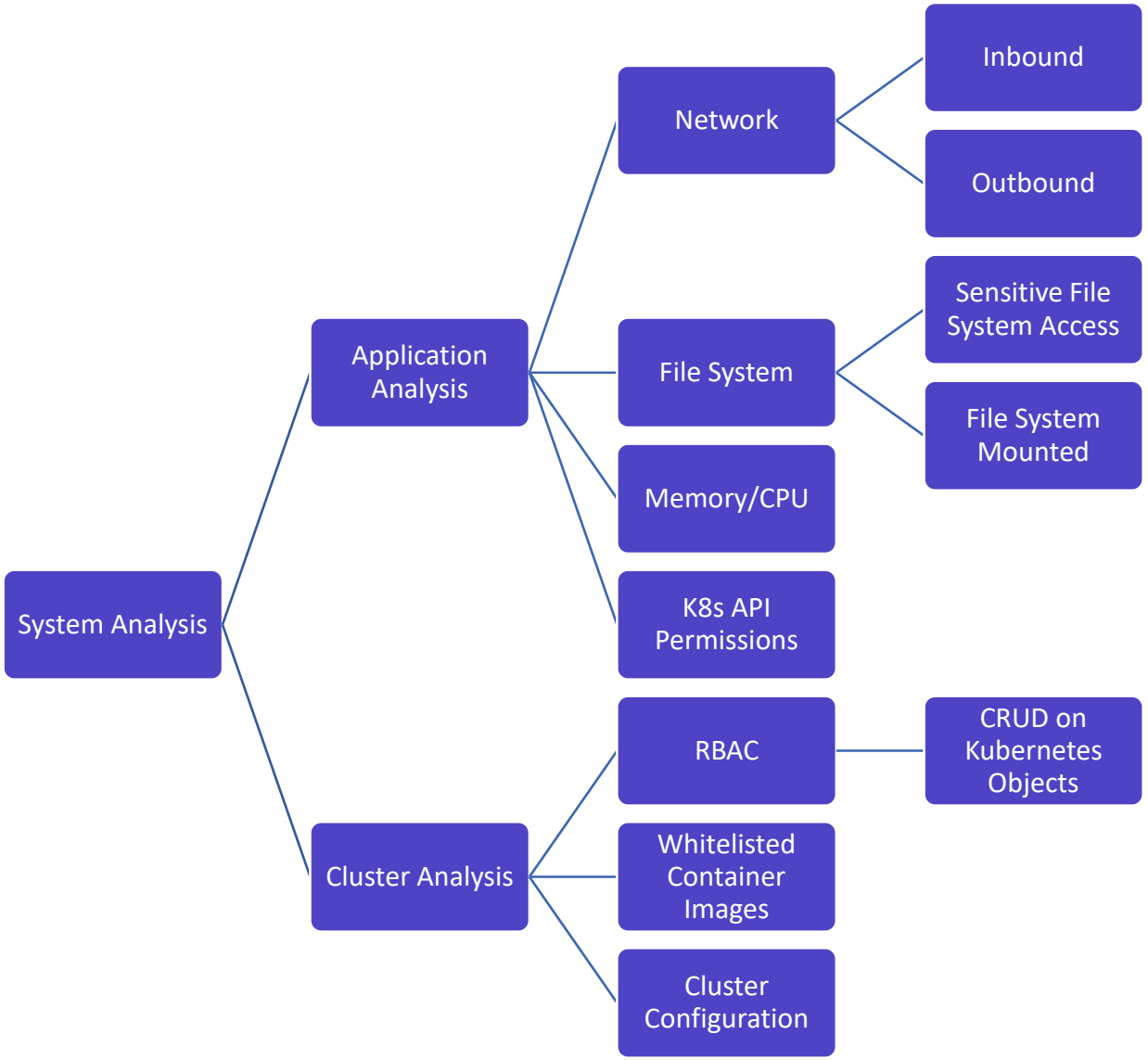
Our Falco Strategy



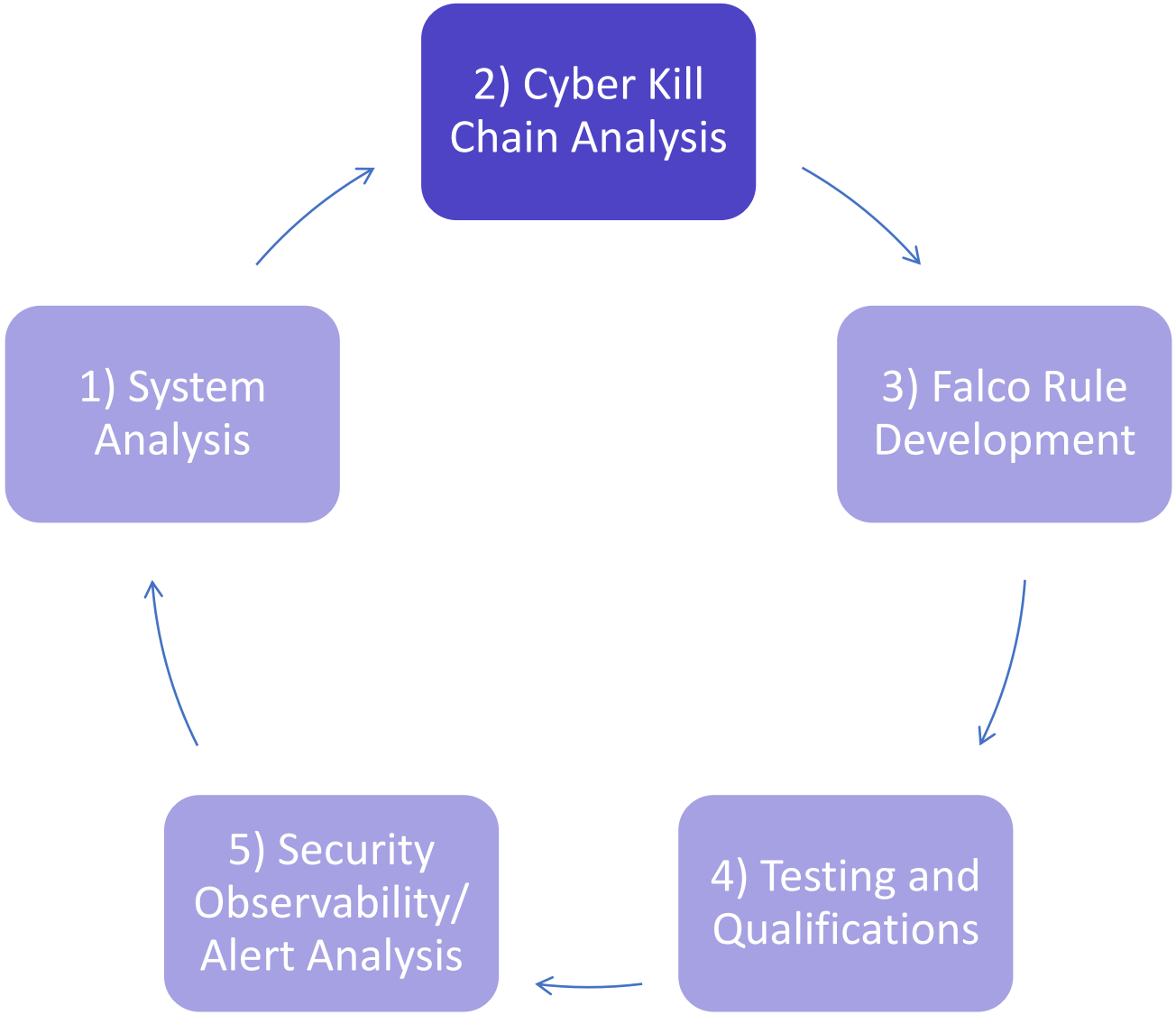
Our Approach To Use Falco



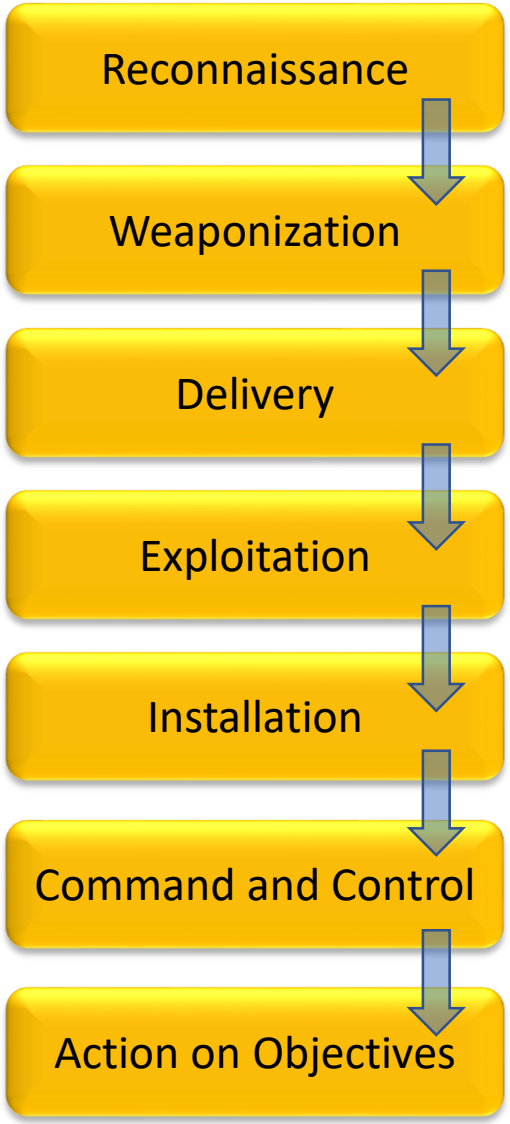
System Analysis



Our Approach To Use Falco

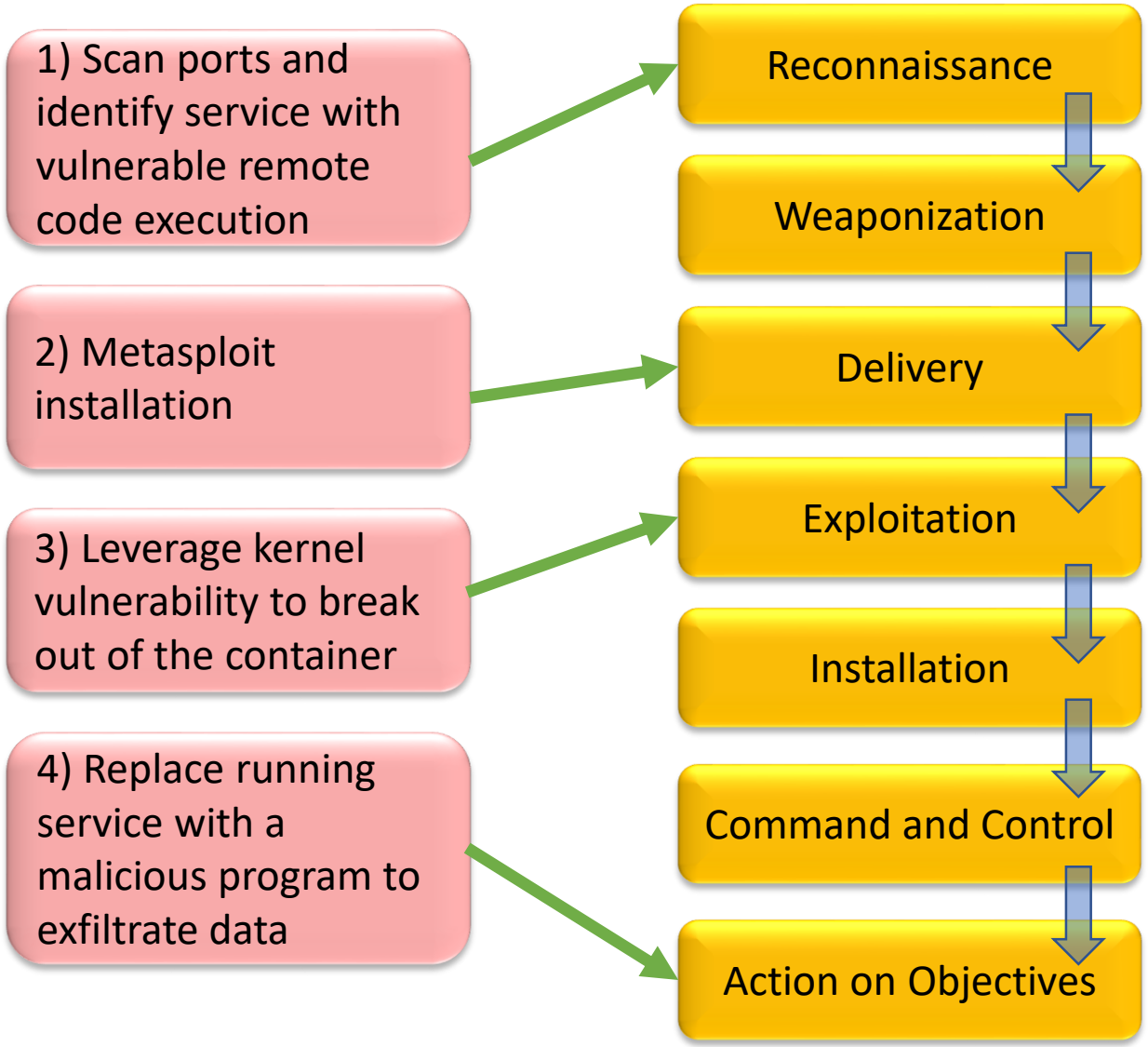


Security Kill Chain with Falco



The Cyber Kill Chain®

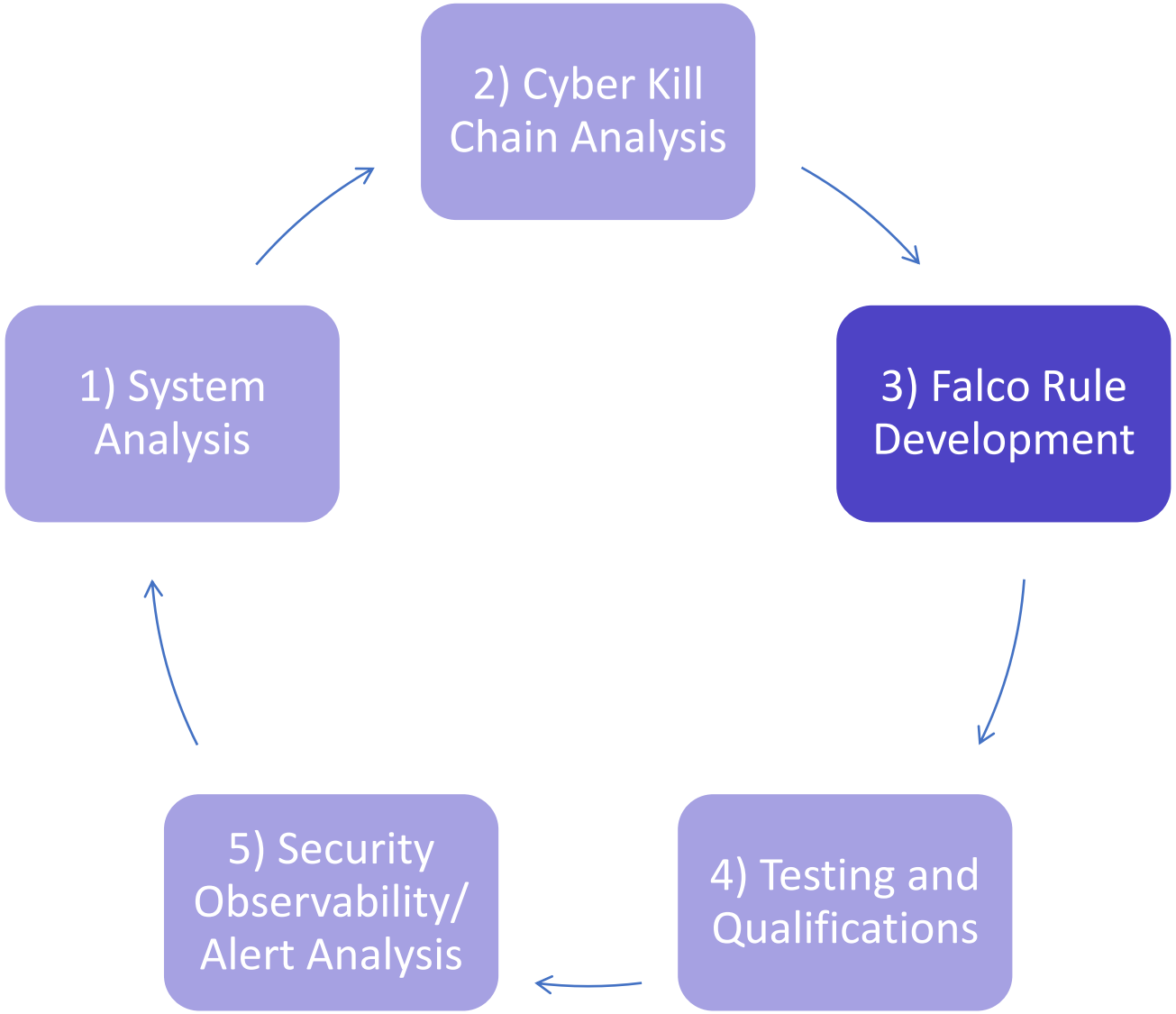
Security Kill Chain



falco

The Cyber Kill Chain®

Our Approach To Use Falco



Rule Example

Unexpected Traffic from SSH Port Scanning

1) Scan ports and identify service with vulnerable remote code execution

```
- macro: allowed_ssh_hosts
  condition: fd.net = “10.97.0.0/24"

- rule: Unsanctioned SSH Connection
  desc: Detect any new ssh connection to a host other than those in an
allowed group of hosts
  condition: (inbound_outbound) and ssh_port and not allowed_ssh_hosts
  output: Unsanctioned SSH Connection (commandLine=%proc.cmdline
connectionId=%fd.name userName=%user.name
containerName=%container.name)
  priority: WARNING
  tags: [network]
```

Rule Example

Unexpected API Server Traffic

- macro: user_known_contact_k8s_api_server_activities
condition: > (container.image.repository in (com.mathworks.webapp-one, com.mathworks.webapp-two))

```
- rule: Contact K8S API Server From Container
  desc: Detect attempts to contact the K8S API Server from a container
  condition: >
    evt.type=connect and evt.dir=< and
    (fd.typechar=4 or fd.typechar=6) and
    container and
    not k8s_containers and
    k8s_api_server and
    not user_known_contact_k8s_api_server_activities
  output: Unexpected connection to K8s API Server from container
(command=%proc.cmdline %container.info
image=%container.image.repository:%container.image.tag connection=%fd.name)
  priority: NOTICE
  tags: [network, k8s, container, mitre_discovery]
```

1) Scan ports and identify service with vulnerable remote code execution

Rule Example

Unexpected Package Installation

2) Metasploit installation

```
# Example (Default Rules Provided by Falco)

- rule: Launch Package Management Process in Container
  desc: Package management process ran inside container
  condition: >
    spawned_process
    and container
    and user.name != "_apt"
    and package_mgmt_procs
    and not package_mgmt_ancestor_procs
    and not user_known_package_manager_in_container
  output: >
    Package management process launched in container (user=%user.name
user_loginuid=%user.loginuid
    command=%proc.cmdline container_id=%container.id container_name=%container.name
image=%container.image.repository:%container.image.tag)
  priority: ERROR
  tags: [process, mitre_persistence]
```

Rule Example

Create symlink over sensitive files

```
# Example (Default Rules Provided by Falco)

- rule: Create Symlink Over Sensitive Files
  desc: Detect symlink created over sensitive files
  condition: >
    create_symlink and
    (evt.arg.target in (sensitive_file_names) or evt.arg.target in
(sensitive_directory_names))
  output: >
    Symlinks created over senstivie files (user=%user.name
user_loginuid=%user.loginuid command=%proc.cmdline
target=%evt.arg.target linkpath=%evt.arg.linkpath
parent_process=%proc.pname)
  priority: NOTICE
  tags: [file, mitre_exfiltration]
```

3) Leverage kernel vulnerability to break out of the container

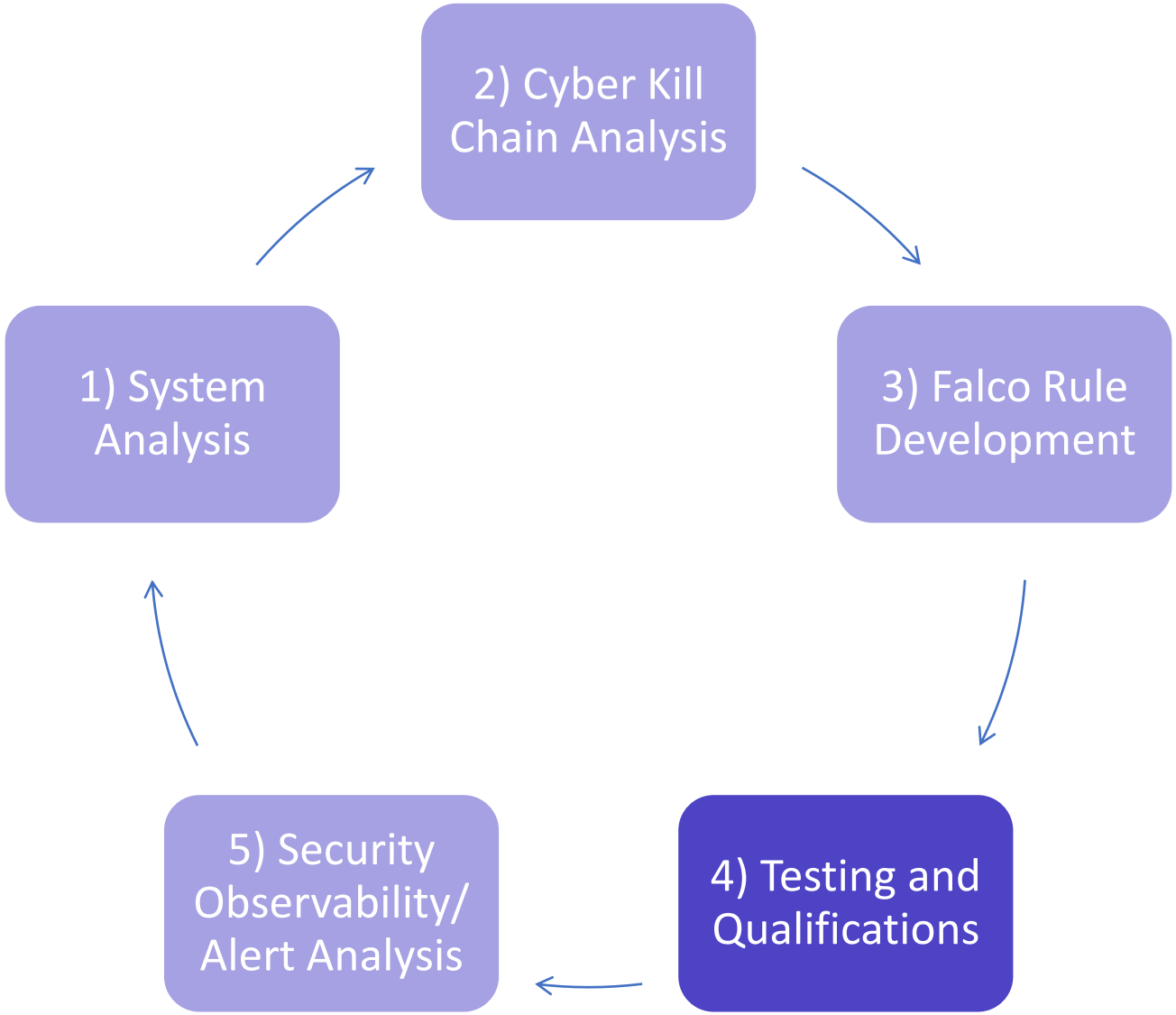
Rule Example

K8s deployment deleted

4) Replace running service with a malicious program to exfiltrate data

```
- rule: K8s Deployment Deleted
  desc: Detect any attempt to delete a deployment
  condition: (kactivity and kdelete and deployment and response_successful)
  output: K8s Deployment Deleted (user=%ka.user.name deployment=%ka.target.name
ns=%ka.target.namespace resp=%ka.response.code decision=%ka.auth.decision
reason=%ka.auth.reason)
  priority: INFO
  source: k8s_audit
  tags: [k8s]
```

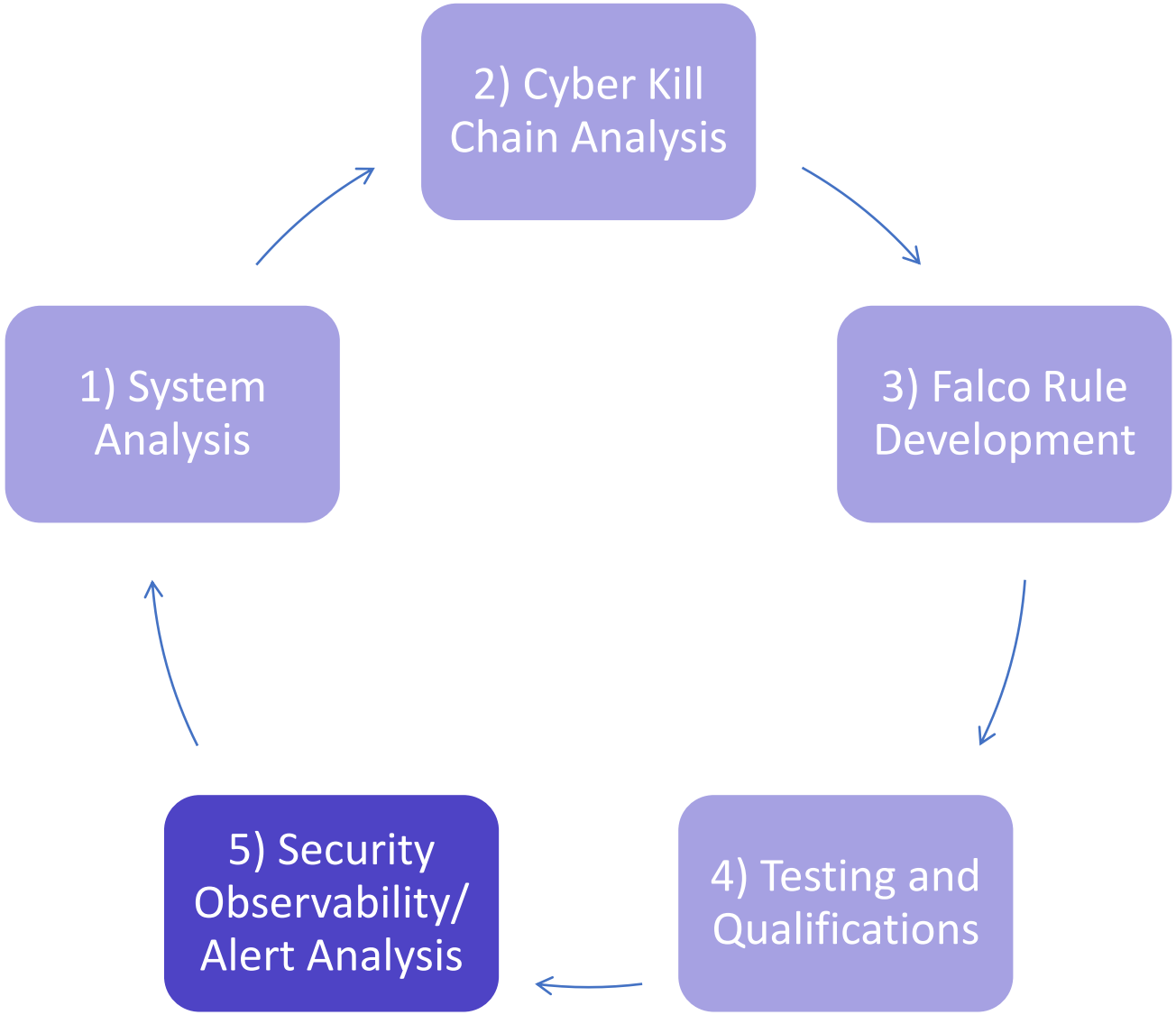
Our Approach To Use Falco



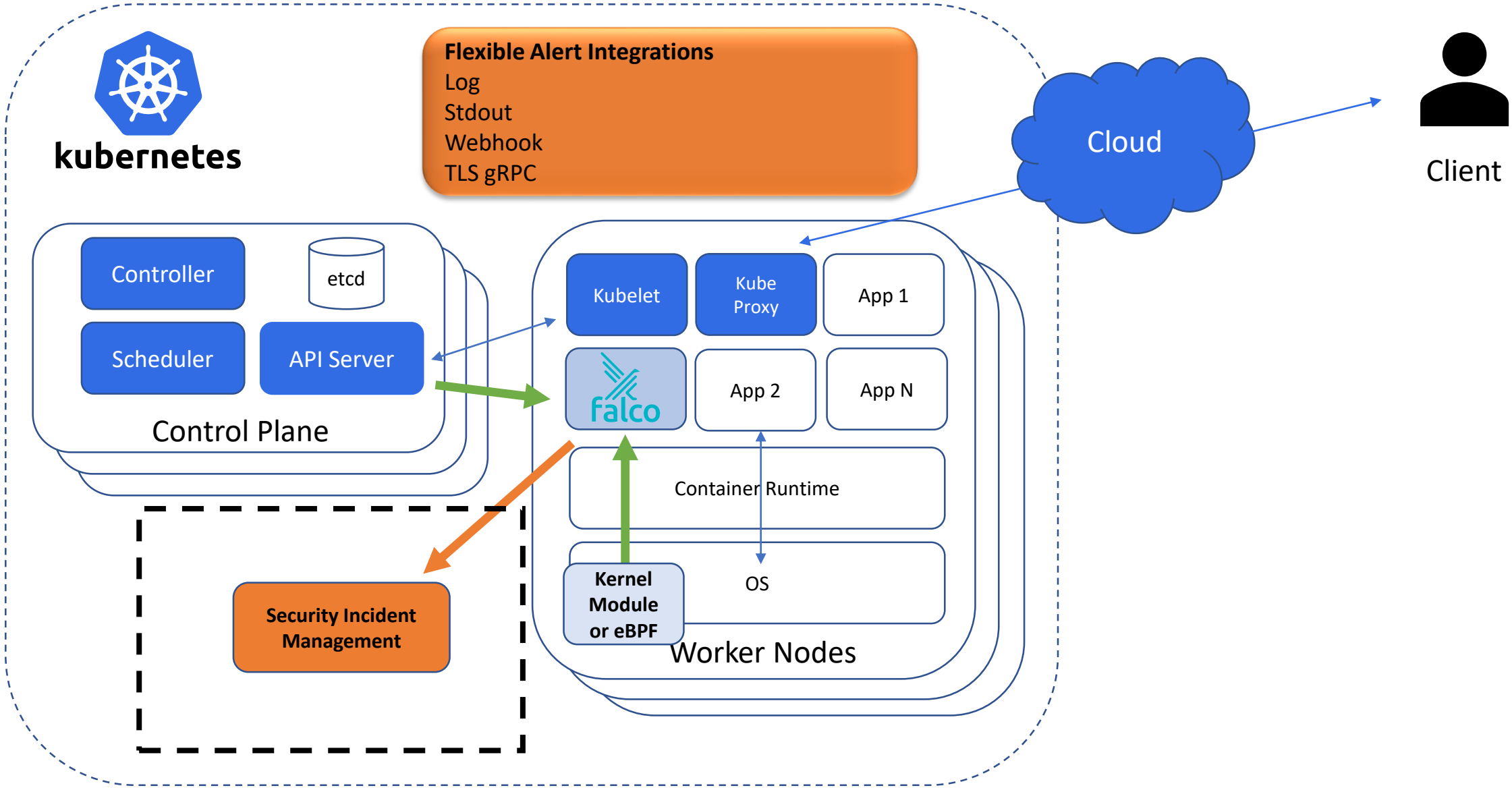
Falco Rules Testing (Demo)

- **Staging Areas**
  - Stage Falco rules in dev environment before production release
- **Strategies**
  1. Manual Testing
  2. Falco Event Generator https://github.com/falcosecurity/event-generator
     - Generate suspect actions (ex. System and Kubernetes Actions)
     - Benchmark Falco

Our Approach To Use Falco



Using Falco Alerts



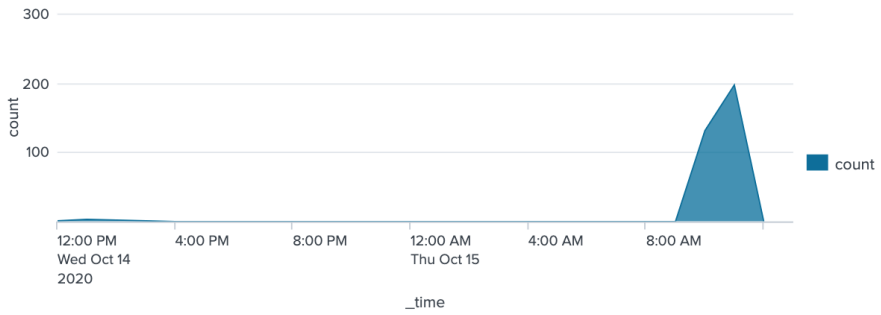
Security Observability

KubeCon CloudNativeCon North America 2020 Virtual

KubeCon CNCF Demo

Edit | Export | ...

File System Integrity

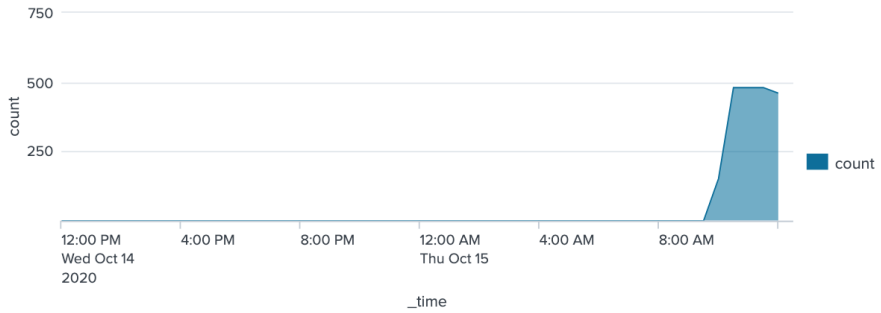


File System Violation Details

command	path	container_id
event-generator	/bin/true	3b48600a8ea5056c54e42636cbce452a10efd6df8fed3e1bbc22b787da49755f
event-generator	/bin/true.event-generator	3b48600a8ea5056c54e42636cbce452a10efd6df8fed3e1bbc22b787da49755f

Unsanctioned Network Traffic

Unexpected Traffic to K8s API Server



Network Traffic Violation Details

container	connection	count
d7f63ff2a333	10.97.136.75:32768->10.96.0.1:443	1
d7f63ff2a333	10.97.136.75:32770->10.96.0.1:443	1
d7f63ff2a333	10.97.136.75:32774->10.96.0.1:443	1
d7f63ff2a333	10.97.136.75:32776->10.96.0.1:443	1
d7f63ff2a333	10.97.136.75:32822->10.96.0.1:443	1
d7f63ff2a333	10.97.136.75:32824->10.96.0.1:443	1
d7f63ff2a333	10.97.136.75:32826->10.96.0.1:443	1
d7f63ff2a333	10.97.136.75:32828->10.96.0.1:443	1

Security Observability

KubeCon CloudNativeCon North America 2020 Virtual

Splunk Alert: System Command Outlier

S @mathworks.com

To Eric Hollis;

Reply | Reply All | Forward | ...

Tue 3/17/2020 12:01 PM

If there are problems with how this message is displayed, click here to view it in a web browser.

The alert condition for System Command Outlier' was triggered.

Alert: System Command Outlier

View results in Splunk

count	containerId	isOutlier	lowerBound	upperBound	avg	stdev
14	e29f1b62f8cd	1	-2.3329827963642920	13.595439617163054	1.6491228070175439	1.9910528016909181

If you believe you've received this email in error, please see your Splunk administrator.

splunk > the engine for machine data


KEEP CLOUD NATIVE
EVERYWHERE
KubeCon
CloudNativeCon
North America 2020
Virtual
HELM
KV