



In Falco's Nest: The Evolution of Cloud Native Runtime Security

Aldo Lacuku, Kong
Iacopo Rozzo, Sysdig

Tuesday March 24, 2026 12:00 - 12:30 CET (G102-103)



KubeCon



CloudNativeCon

Europe 2026

Meet the maintainers



Aldo Lacuku

 [alacuku](#)

@Kong Inc.



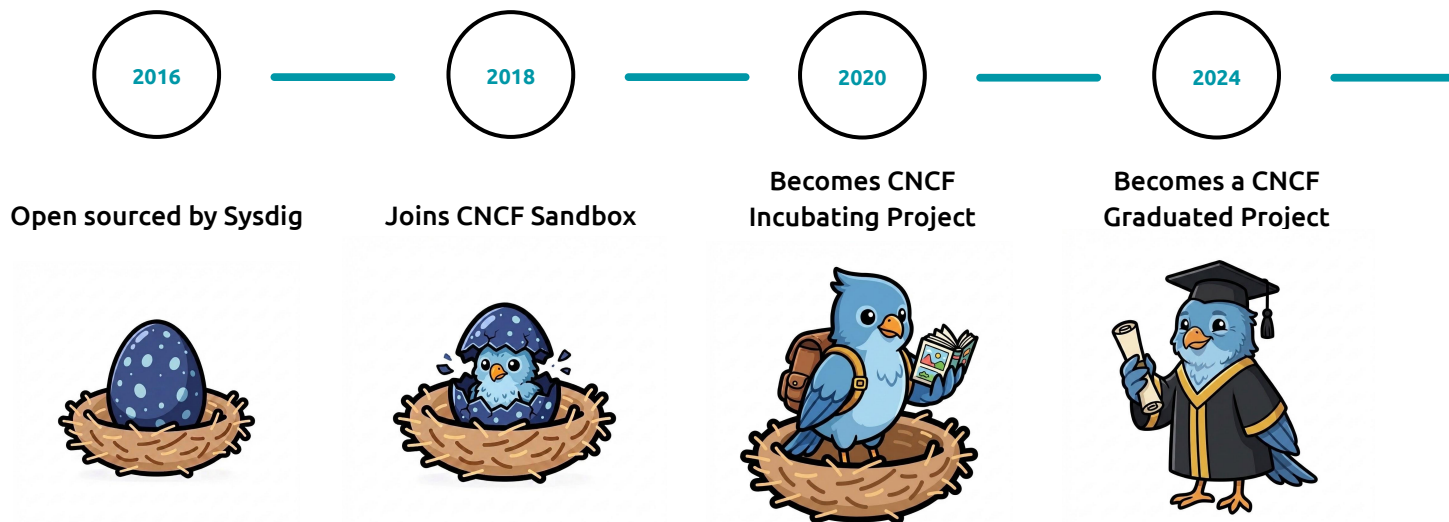
Iacopo Rozzo

 [irozzo-1A](#)

@Sysdig

A Decade of Detection 🎉

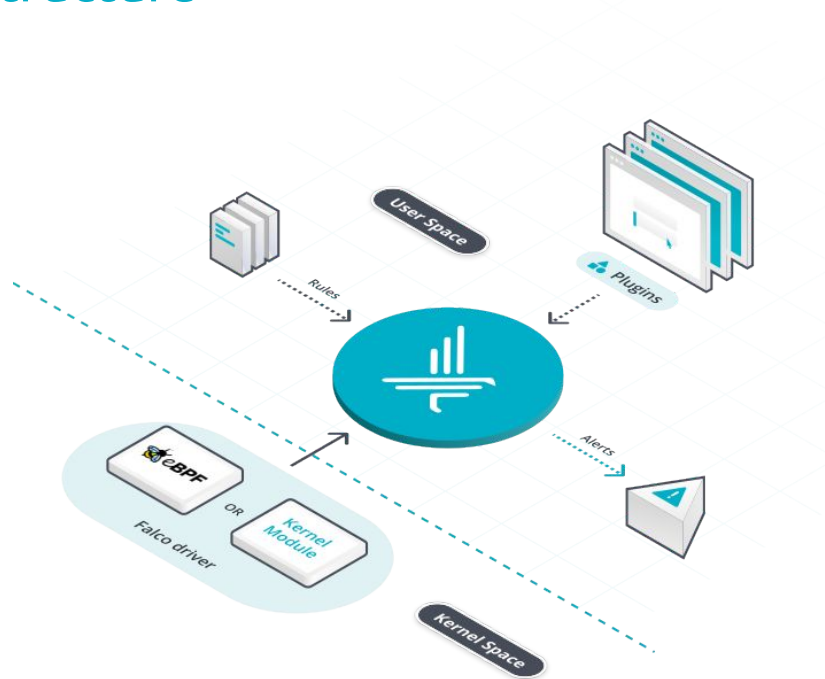
From a kernel-tapping experiment to a global standard



What is Falco?

The security cameras for your infrastructure

- Detect security threats in real time
- Kernel level visibility
- Highly extensible
- Cloud native





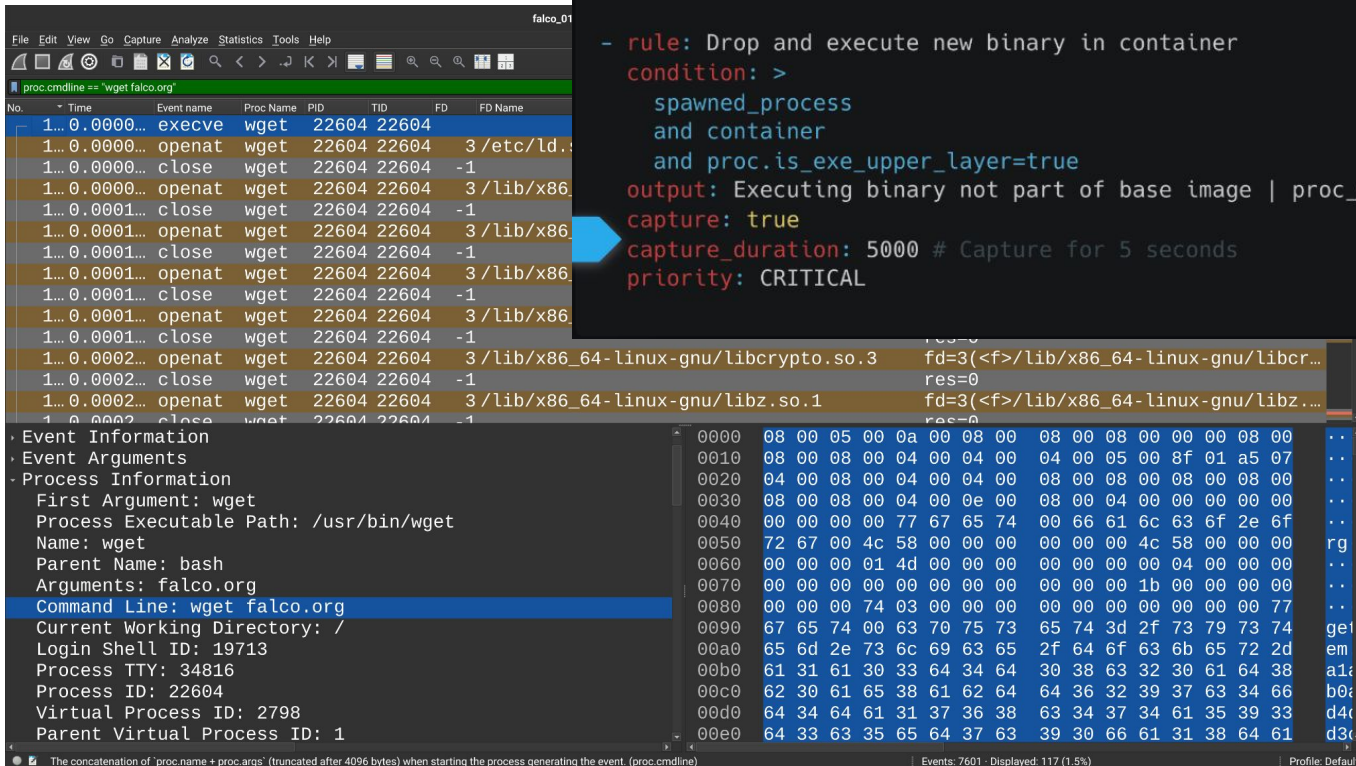
What's New

Falco 0.42 & 0.43



Capture recording / Stratoshark

Forensics with Falco



The screenshot shows the Falco GUI interface. The top panel displays a list of events captured by Falco. The bottom panel shows a detailed view of a specific event, including process information and arguments.

Event List:

No.	Time	Event name	Proc Name	PID	TID	FD	FD Name
1...	0.0000...	execve	wget	22604	22604		
1...	0.0000...	openat	wget	22604	22604	3	/etc/ld.so
1...	0.0000...	close	wget	22604	22604	-1	
1...	0.0000...	openat	wget	22604	22604	3	/lib/x86_64-linux-gnu/libcrypto.so.3
1...	0.0001...	close	wget	22604	22604	-1	
1...	0.0001...	openat	wget	22604	22604	3	/lib/x86_64-linux-gnu/libz.so.1
1...	0.0001...	close	wget	22604	22604	-1	
1...	0.0001...	openat	wget	22604	22604	3	/lib/x86_64-linux-gnu/libcrypto.so.3
1...	0.0001...	close	wget	22604	22604	-1	
1...	0.0001...	openat	wget	22604	22604	3	/lib/x86_64-linux-gnu/libz.so.1
1...	0.0001...	close	wget	22604	22604	-1	
1...	0.0002...	openat	wget	22604	22604	3	/lib/x86_64-linux-gnu/libcrypto.so.3
1...	0.0002...	close	wget	22604	22604	-1	
1...	0.0002...	openat	wget	22604	22604	3	/lib/x86_64-linux-gnu/libz.so.1
1...	0.0002...	close	wget	22604	22604	-1	

Event Information:

- Event Arguments
- Process Information
 - First Argument: wget
 - Process Executable Path: /usr/bin/wget
 - Name: wget
 - Parent Name: bash
 - Arguments: falco.org
 - Command Line: wget falco.org
 - Current Working Directory: /
 - Login Shell ID: 19713
 - Process TTY: 34816
 - Process ID: 22604
 - Virtual Process ID: 2798
 - Parent Virtual Process ID: 1

Rule Configuration:

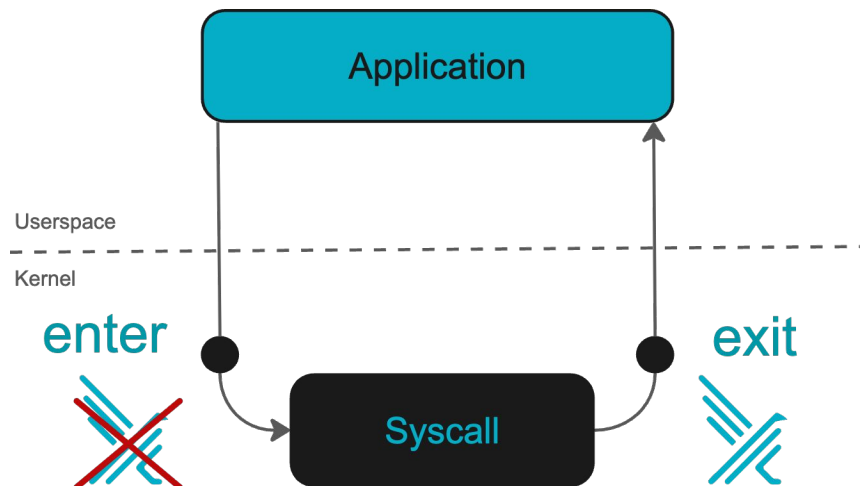
```

- rule: Drop and execute new binary in container
  condition: >
    spawned_process
    and container
    and proc.is_exe_upper_layer=true
  output: Executing binary not part of base image | proc_exe=%proc.exe ...
  capture: true
  capture_duration: 5000 # Capture for 5 seconds
  priority: CRITICAL
  
```

Drop enter initiative

Performance matters

- Lighter Kernel instrumentation (up to 20% latency reduction)
- Kernel events reduced by 1/2 (up to 30% less Falco CPU usage)
- **evt.dir** now deprecated



Feathering the Nest

Other Key Changes

- Falco 0.42

- Plugin event schema versioning
- Thread table auto-purging configuration
- Static fields

- Falco 0.43

- Container plugin performance and stability improvements
- Deprecated gRPC output and gVisor engine
- Scheduled maintenance: GPG key rotation



Falco Operator

Kubernetes-Native Deployment and Management for Falco





Falco Operator

v0.2.0

Kubernetes-native deployment and management for Falco

5 Custom Resources

Improved OCI Artifacts

ConfigMap Support

Ecosystem Components

Conditions and Events



Falco Instance

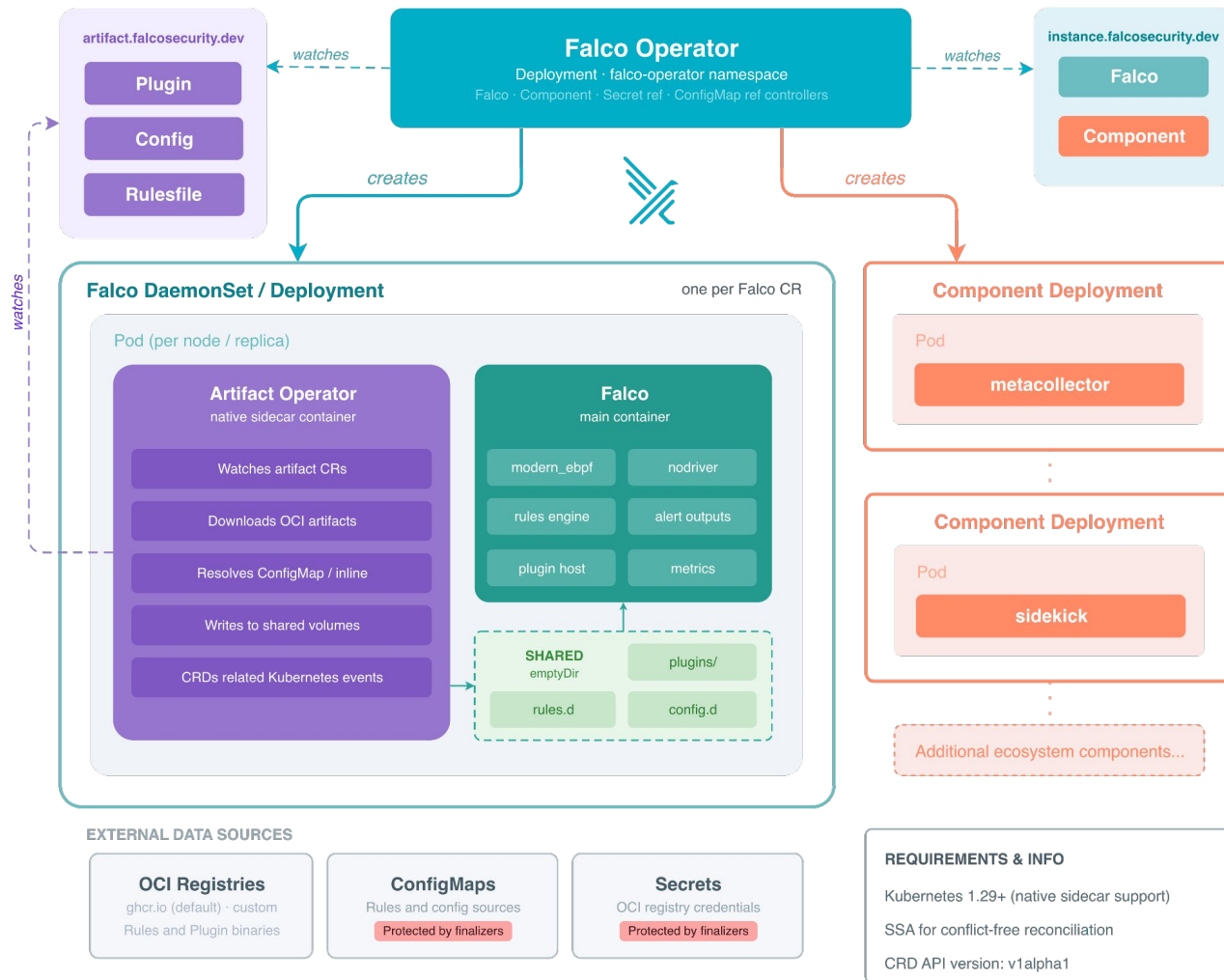
falco.org

metacollector · sidekick · sidekick-ui



Architecture Overview





Major Features and Improvements

Structured API types

Before (0.1.x)

```
1 apiVersion: artifact.falcosecurity.dev/v1alpha1
2 kind: Config
3 metadata:
4   name: my-config
5 spec:
6   config: |-
7     engine:
8       kind: modern_ebpf
9     output_timeout: 2000
```

After (0.2.0)

```
1 apiVersion: artifact.falcosecurity.dev/v1alpha1
2 kind: Config
3 metadata:
4   name: my-config
5 spec:
6   config:
7     engine:
8       kind: modern_ebpf
9     output_timeout: 2000
```

Redesigned OCI Artifact API

Before (0.1.x)

```
1 apiVersion: artifact.falcosecurity.dev/v1alpha1
2 kind: Rulesfile
3 metadata:
4   name: falco-rules
5 spec:
6   ociArtifact:
7     reference: ghcr.io/falcosecurity/rules/falco-rules:latest
8     pullSecret:
9       secretName: my-secret
```

After (0.2.0)

```
1 apiVersion: artifact.falcosecurity.dev/v1alpha1
2 kind: Rulesfile
3 metadata:
4   name: falco-rules
5 spec:
6   ociArtifact:
7     image:
8       repository: falcosecurity/rules/falco-rules
9       tag: latest
10    registry:
11      name: ghcr.io
12      auth:
13        secretRef:
14          name: my-secret
```

Configmap Support

Rulesfiles

```
1 apiVersion: artifact.falcosecurity.dev/v1alpha1
2 kind: Rulesfile
3 metadata:
4   name: custom-rules
5 spec:
6   configMapRef:
7     name: my-rules-configmap
8   priority: 50
```

Configs

```
1 apiVersion: artifact.falcosecurity.dev/v1alpha1
2 kind: Config
3 metadata:
4   name: custom-config
5 spec:
6   configMapRef:
7     name: my-config-configmap
8   priority: 50
```



Ecosystem Components

```
1 apiVersion: instance.falcosecurity.dev/v1alpha1
2 kind: Component
3 metadata:
4   name: falcosidekick
5 spec:
6   component:
7     type: falcosidekick
8   replicas: 2
9   podTemplateSpec:
10    spec:
11     containers:
12     - name: falcosidekick
13       env:
14       - name: WEBUI_URL
15         value: "http://falcosidekick-ui.falco.svc:2802"
```

TYPE: METACOLLECTOR

k8s-metacollector

Centralized Kubernetes metadata enrichment for Falco instances

Kubernetes metadata

TYPE: FALCOSIDEKICK

Falcosidekick

Routes Falco events to 70+ integrations: Slack, S3, Kafka and more

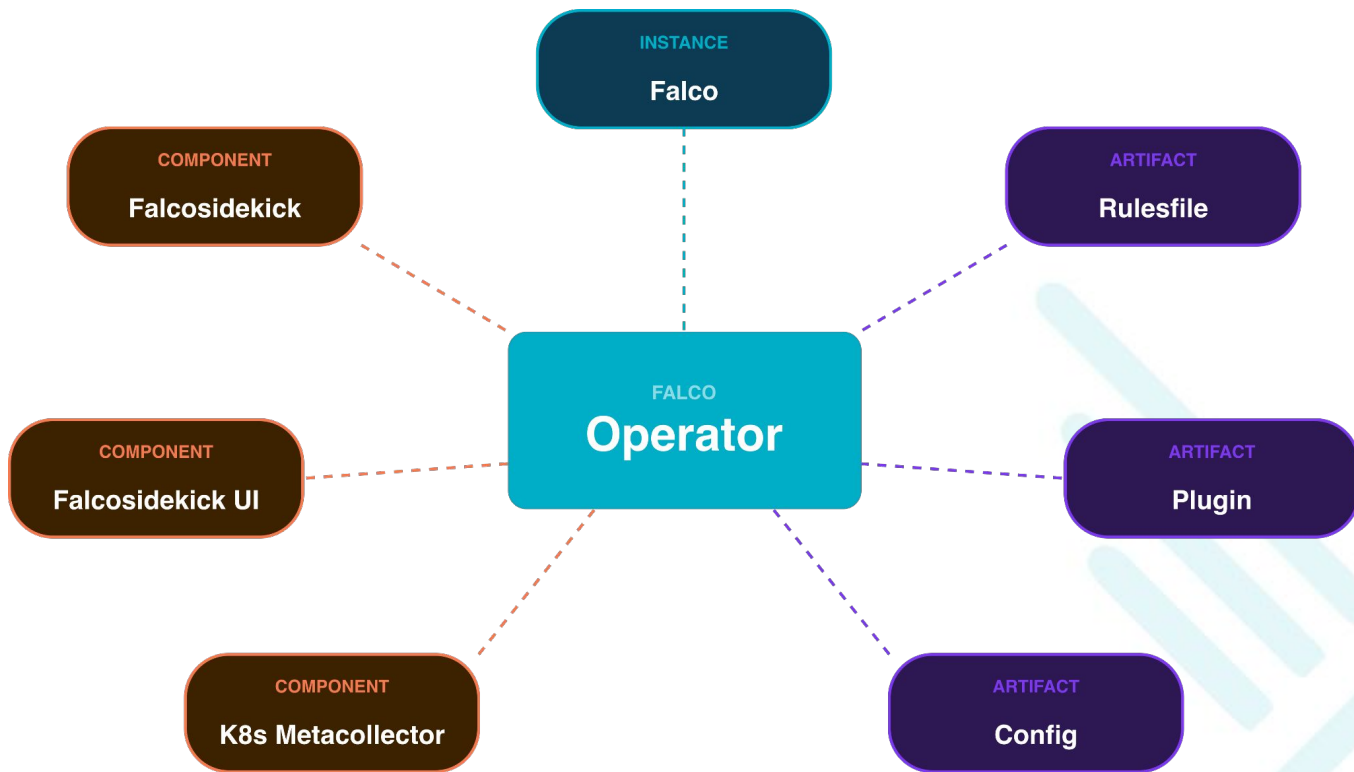
70+ integrations

TYPE: FALCOSIDEKICK-UI

Falcosidekick UI

Web dashboard for real-time Falco event visualization

Requires Redis





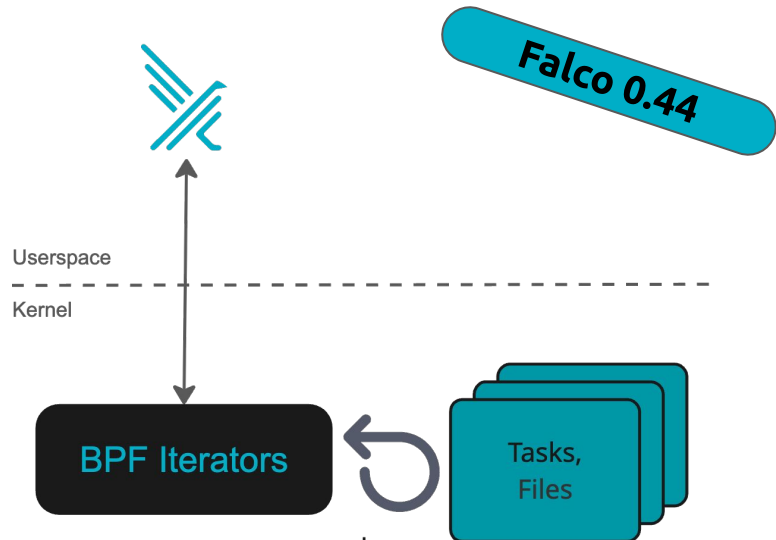
What's Next

Upcoming in Falco 0.44 and beyond



BPF Iterators

- Improve information gathering performance over *procf*s scan
- Used for initialization of thread information and state recovery



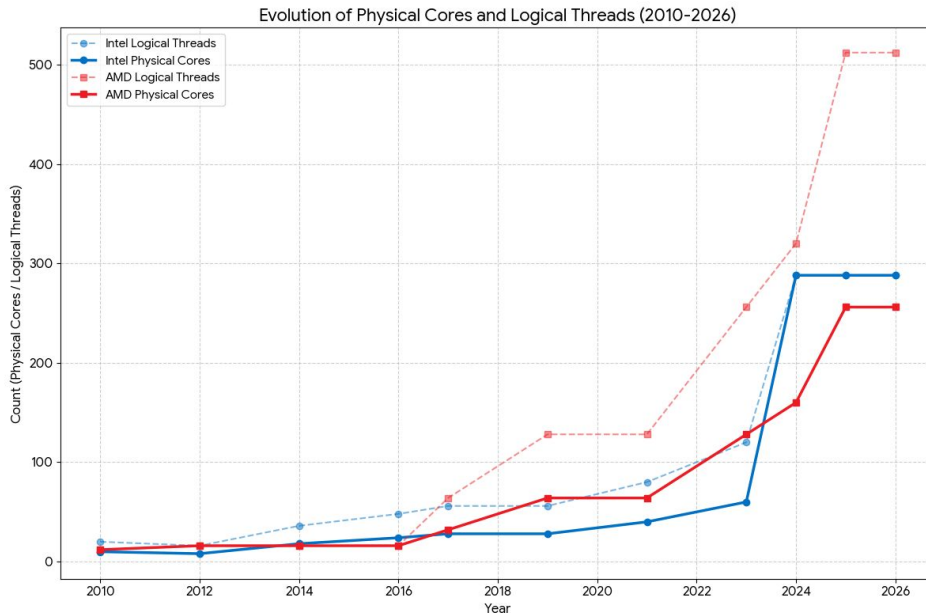
Falco 0.44

```
struct bpf_iter_task_file {
    union {
        struct bpf_iter_meta
*meta;
    };
    union {
        struct task_struct *task;
    };
    u32 fd;
    union {
        struct file *file;
    };
};
```

Multi-thread Support

Why do we need this?

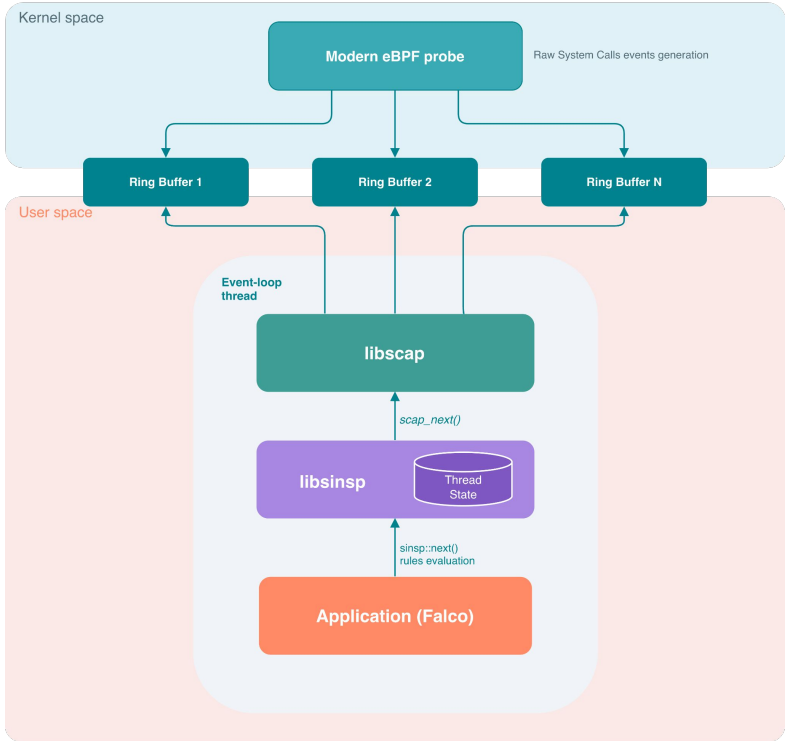
- ✓ Falco single-threaded event loop architecture, simple and blazingly fast
- ✗ Single event loop risks to be increasingly subject to saturation on high-core machines



Soon in Tech Preview

Multi-thread Support

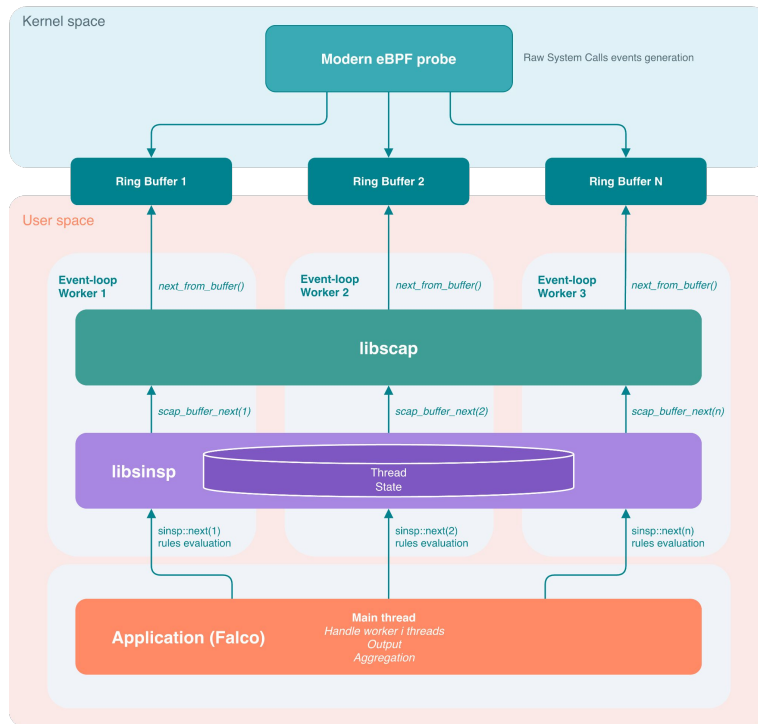
Current single-threaded architecture



- ### Event Processing Flow
1. Kernel captures system call
 2. Event stored in per-CPU ring buffer
 3. `scap_next()` reads next from buffer
 4. `sinsp::next()` processes event
 5. `sinsp_parser::process_event()`
 6. Thread state updated
 7. Event returned to Application

Multi-thread Support

Proposed multi-threaded architecture



Event Processing Flow

1. Kernel captures system call
2. Event stored in per-TID ring buffer
3. `scap_buffer_next()` reads next from buffer (each worker thread)
4. `sinsp::next()` processes event (per worker thread)
5. `sinsp::process_event()` (per worker thread)
6. Thread state updated (per worker thread)
7. Rules evaluation (per worker thread)



What's Next

The Developer Experience



Falco LSP

Language Tooling for Falco Rules

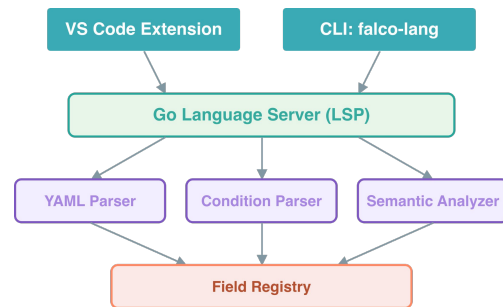
Soon on
VSCode Marketplace!

What it is

- Language Server Protocol implementation + CLI tool + VS Code Extension
- **Real-time feedback** while writing Falco rules directly in your editor!

Key capabilities

- Real-time **diagnostics**: syntax errors, undefined macros/lists, unknown fields, source mismatches
- Context-aware **code completion**: fields, operators, macros, lists, priorities, sources
- **Hover** documentation for fields, macros, and lists
- **Go-to-definition** and find references across files
- **Code formatting** with falco-lang format
- **Validation** in CI pipelines with falco-lang validate



Falco for Coding Agents

Tech Preview

```
leogr@x86 [?] [?] ~/code/github.com/falcosecurity/falco [?] .....  
> |
```



Community updates

We need you!



Contributor funding program

Sysdig Celebrates 10 Years of Falco with \$70,000 Donation

- Feature development grants to speed the delivery of new capabilities
- Contributor stipends to recognize and retain key contributors
- Technical writer stipends to improve both documentation and user experience



Thank you!

Join Our Community and Reach Out to Us!



#falco channel on
kubernetes.slack.com



Community Calls
every 2 weeks on
Wed



cncf-falco-dev@lists.cncf.io