Intrusion Detection with Falco Open Source

Luca Guerra, Sysdig

About Me

- Open Source Engineer at Sysdig
- Falco contributor
- Talk to me about security!
- Worked on Sysdig Secure, OSS Sysdig and Falco
- https://github.com/LucaGuerra
- Kubernetes Slack, #falco

About Falco

Falco, the cloud-native runtime security project, is the de facto *Kubernetes threat detection engine*

Created by

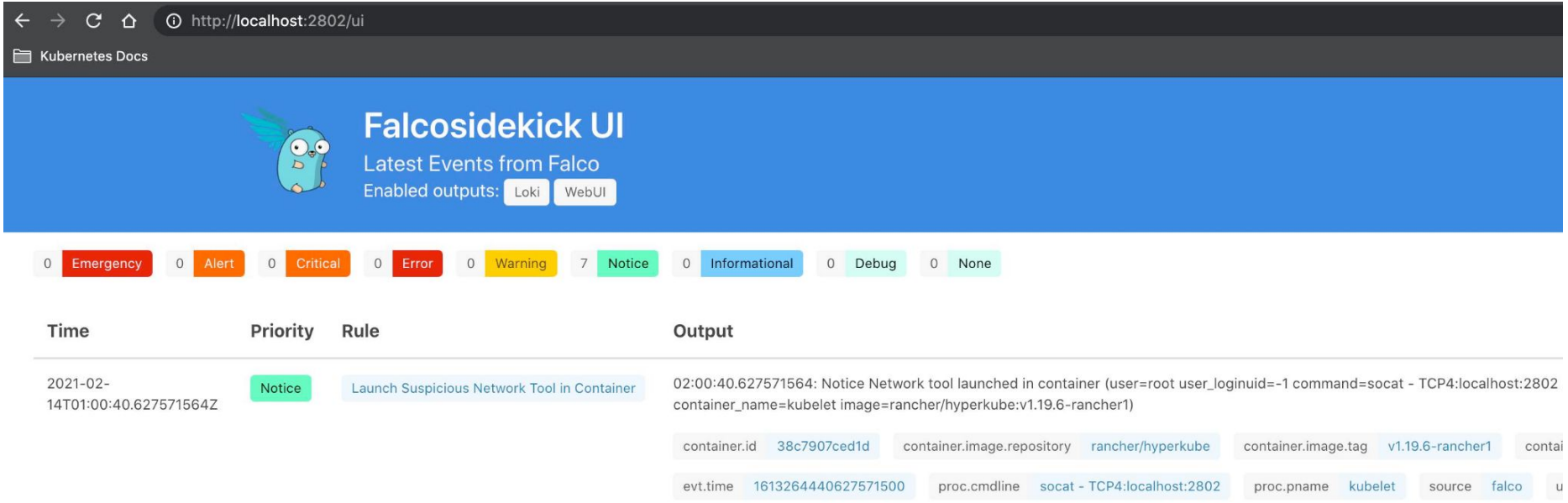
Incubating at

CLOUD NATIVE
COMPUTING FOUNDATION

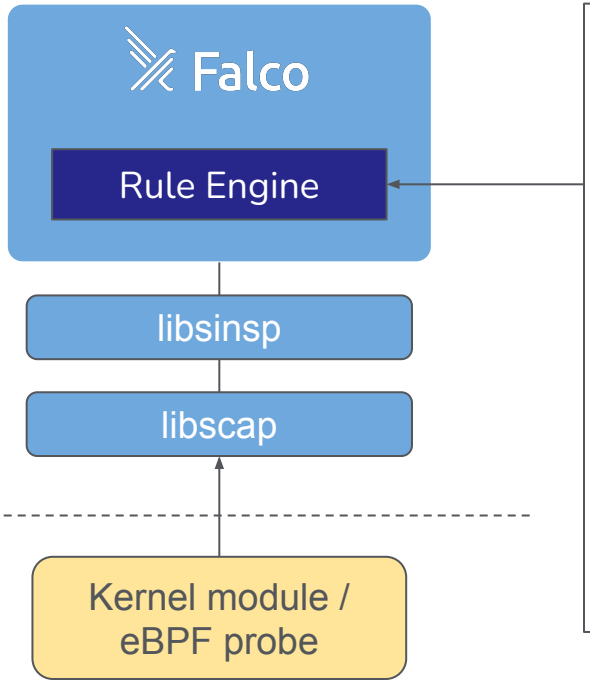
About Falco

```
2022-04-07T12:51:08: Notice A shell was spawned in a container with an attached terminal (user=root
user_loginuid=-1 elastic_borg (id=a10bd3b1b2a8) shell=bash parent=<NA> cmdline=bash terminal=34816
container_id=a10bd3b1b2a8 image=ubuntu)
2022-04-07T12:51:41: Warning Netcat runs inside container that allows remote code execution
(user=root user_loginuid=-1 command=nc -e container_id=a10bd3b1b2a8 container_name=elastic_borg
image=ubuntu:latest)
```

About Falco

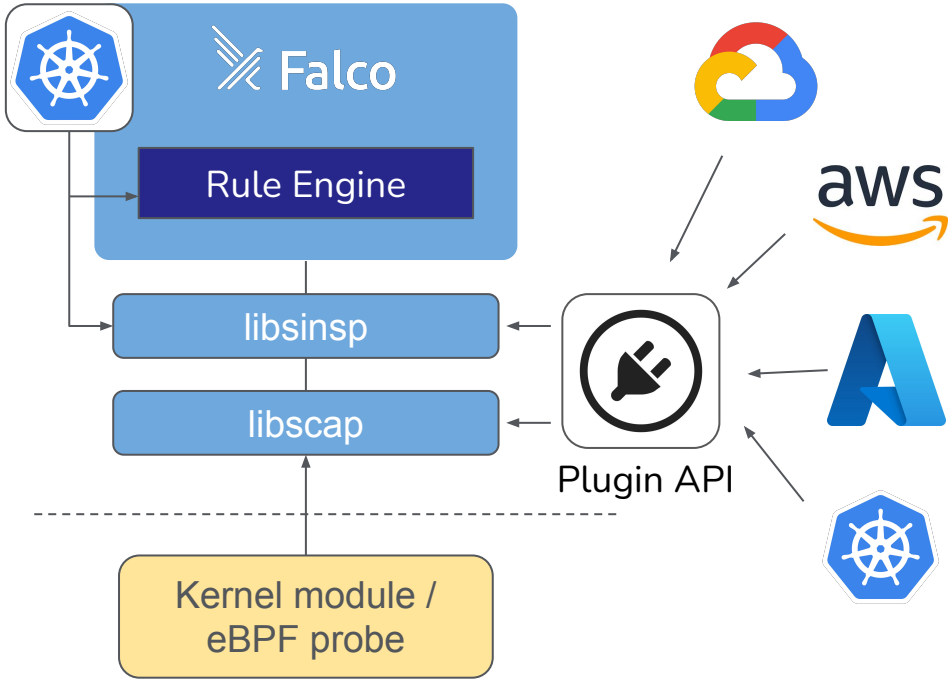

http://localhost:2802/ui
Kubernetes Docs
Falcosidekick UI
Latest Events from Falco
Enabled outputs: Loki WebUI
0 Emergency 0 Alert 0 Critical 0 Error 0 Warning 7 Notice 0 Informational 0 Debug 0 None
Time
Priority
Rule
Output
2021-02-14T01:00:40.627571564Z
Notice
Launch Suspicious Network Tool in Container
02:00:40.627571564: Notice Network tool launched in container (user=root user_loginuid=-1 command=socat - TCP4:localhost:2802 container_name=kubelet image=rancher/hyperkube:v1.19.6-rancher1)
container.id 38c7907ced1d
container.image.repository rancher/hyperkube
container.image.tag v1.19.6-rancher1
evt.time 1613264440627571500
proc.cmdline socat - TCP4:localhost:2802
proc.pname kubelet
source falco

How Falco Works

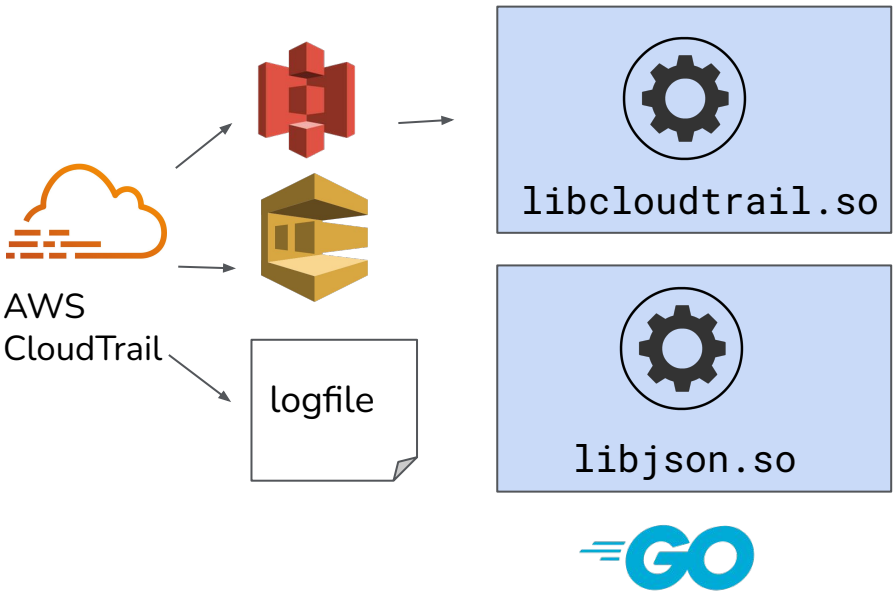



```
- rule: Terminal shell in container
  desc: A shell was used as the entrypoint/exec point into a
container with an attached terminal.
  condition: >
      spawned_process and container
      and shell_procs and proc.tty != 0
      and container_entrypoint
      and not
user_expected_terminal_shell_in_container_conditions
  output: >
      A shell was spawned in a container with an attached
terminal (user=%user.name user_loginuid=%user.loginuid
%container.info
      shell=%proc.name parent=%proc.pname
cmdline=%proc.cmdline terminal=%proc.tty
container_id=%container.id image=%container.image.repository)
  priority: NOTICE
  tags: [container, shell, mitre_execution]
```

How Falco is Evolving



Falco Plugin 💎: AWS CloudTrail




```
plugins:
  - name: cloudtrail
    library_path: libcloudtrail.so
    init_config: ""
    open_params: "s3://..."
  - name: json
    library_path: libjson.so
    init_config: ""

load_plugins: [cloudtrail, json]
```

DEMO

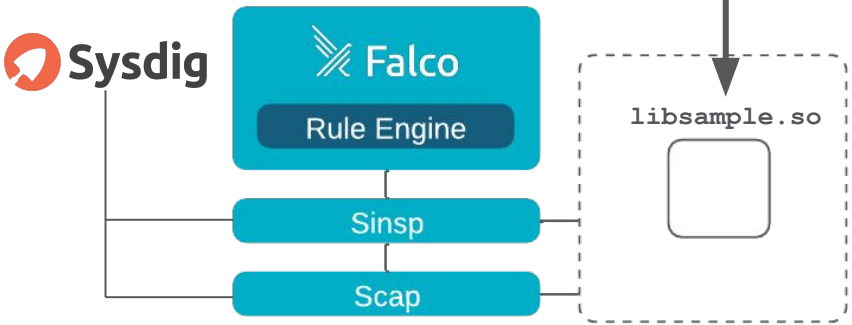
ATTACKER

Bring your own plugins

falco.yaml

```
plugins:
  - name: sample
    library_path: libsample.so
    init_config:
        name: Jason

load_plugins: [sample]
```



sampleplugin.go

```
type PluginConfig struct {
      Name string `json:"name"
}

type Plugin struct {
      plugins.BasePlugin
      conf PluginConfig
}

func init() {
      p := &Plugin{}
      extractor.Register(p)
      source.Register(p)
}

func (m *MyPlugin) Init(c string) error {
      return json.Unmarshal([]byte(c), &m.conf)
}

func (m *MyPlugin) Destroy() {
      println("👋 See you ", m.conf.Name)
}
```

Plugin Caveats ⚠️

- Plugins are a brand new feature for Falco and libraries, and so the Falco community is still experimenting a lot with it
- Feedback from users and plugin authors is **very important** and will shape the future of the product!
- Currently Falco is able to run several plugins (one source+many extractor) for a single source at a time
- It is not currently possible to have syscalls + plugin sources enabled at the same time, but this may change in the future

The ecosystem is growing 🚀

https://github.com/falcosecurity/plugins/tree/master/plugins/cloudtrail

https://github.com/falcosecurity/plugins/tree/master/plugins/json

https://github.com/lssif/docker-plugin

https://github.com/kinvolk/seccompagent

https://twitter.com/developerguyba/status/...

https://github.com/falcosecurity/plugins/tree/master/plugins/okta

Luca Guerra


github.com/LucaGuerra

Thank you!