Cloud Native Days Austria
2025

Catch Hackers with Koney

Automated Honeytokens for Cloud-Native Apps

Cloud Native Days Austria 2025

CO-DEVELOPER

Matteo Golinelli

University of Trento

Catch Hackers with Koney

Automated Honeytokens for Cloud-Native Apps

PRESENTER

Mario Kahlhofer

Dynatrace Research

whoami

MARIO KAHLHOFER

Senior Research Scientist · Dynatrace Research

PhD Student · Johannes Kepler University Linz

mario.kahlhofer@dynatrace.com

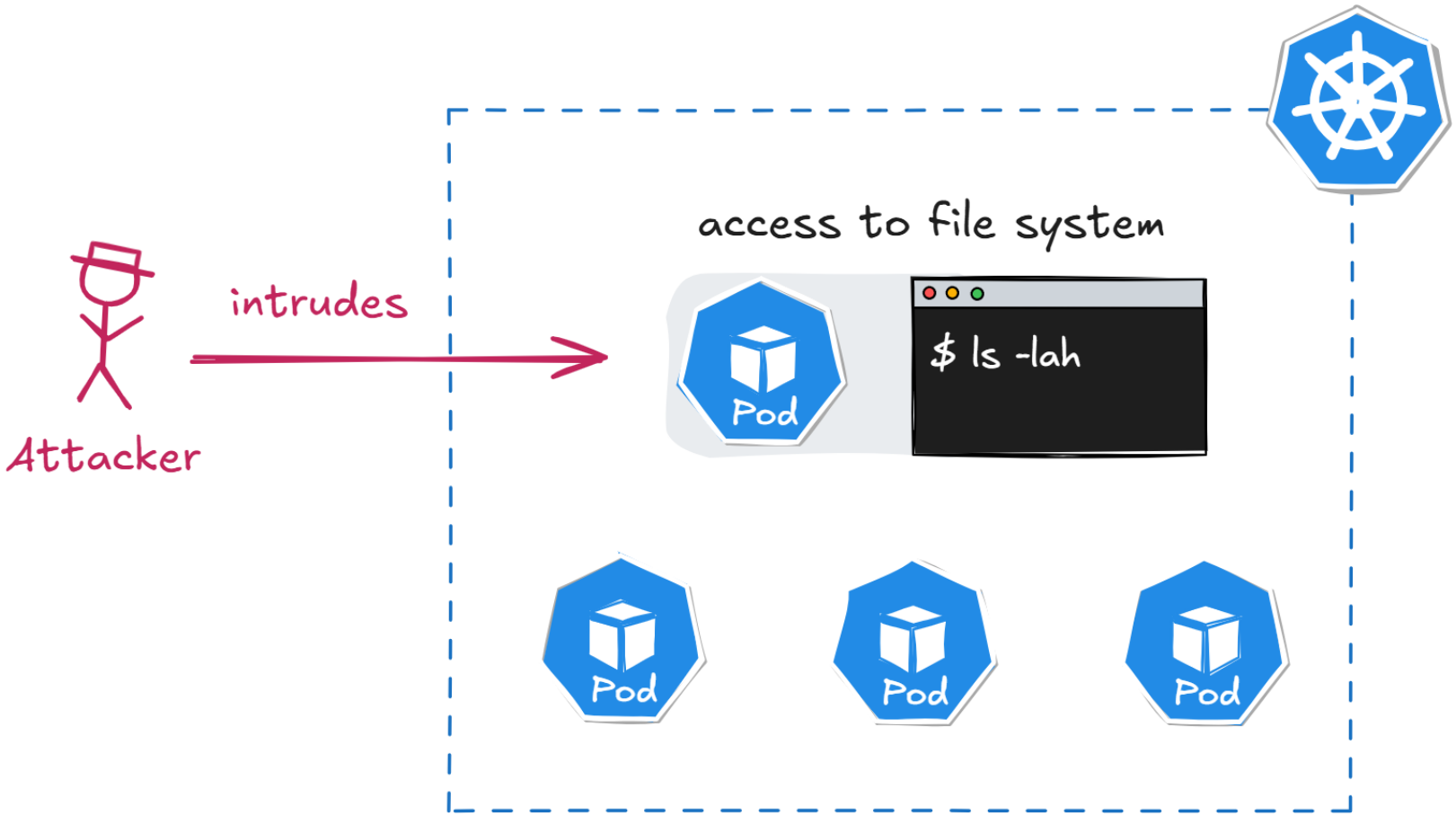
github.com/blu3r4y

LinkedIn

Cloud Security · **Kubernetes** · Containers · Linux · eBPF
Cyber Deception · **Honeytokens** · Honeypots
Game Theory · Machine Learning

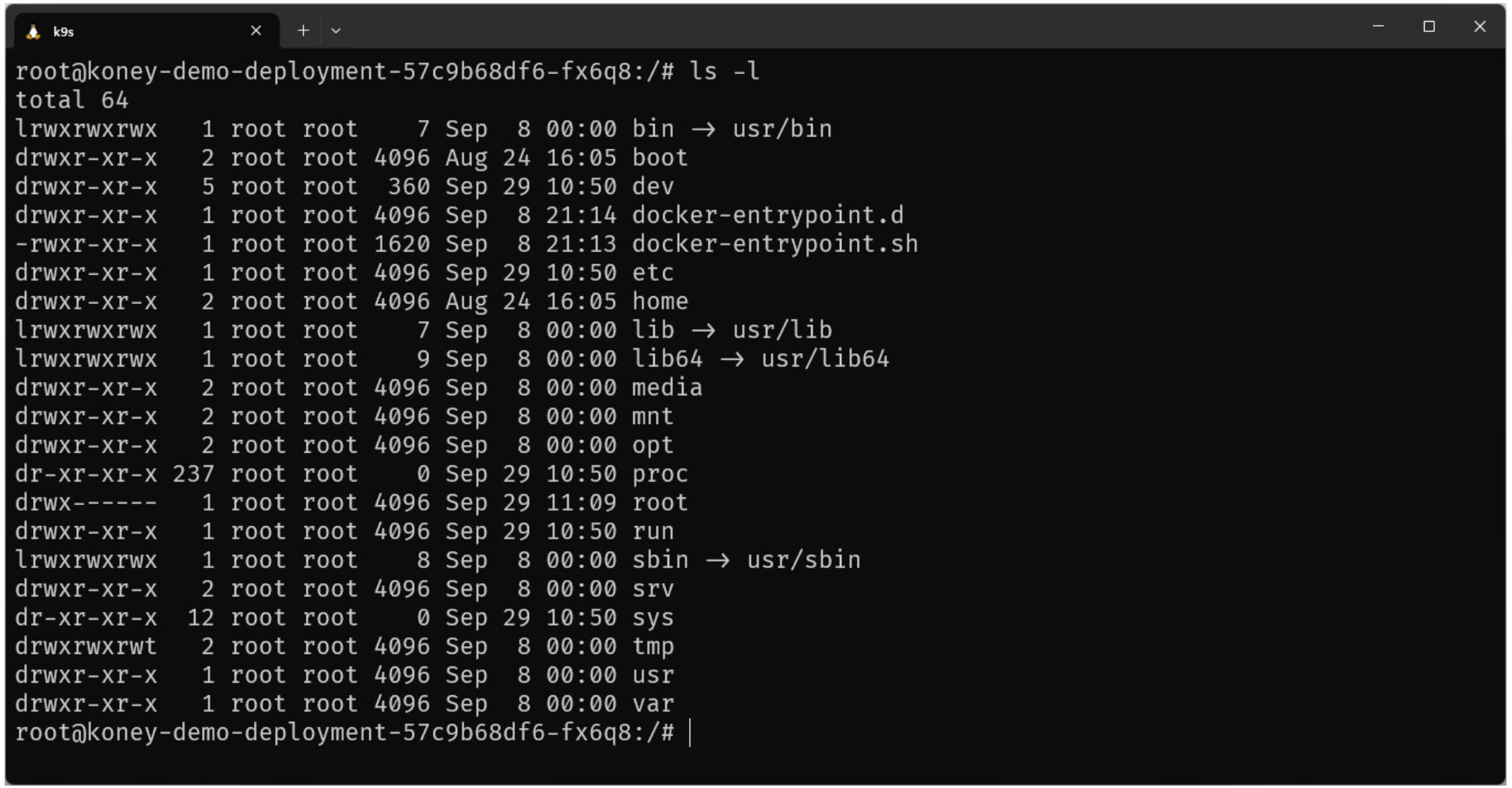
INTRODUCTION

Imagine an attacker infiltrated your cluster ...

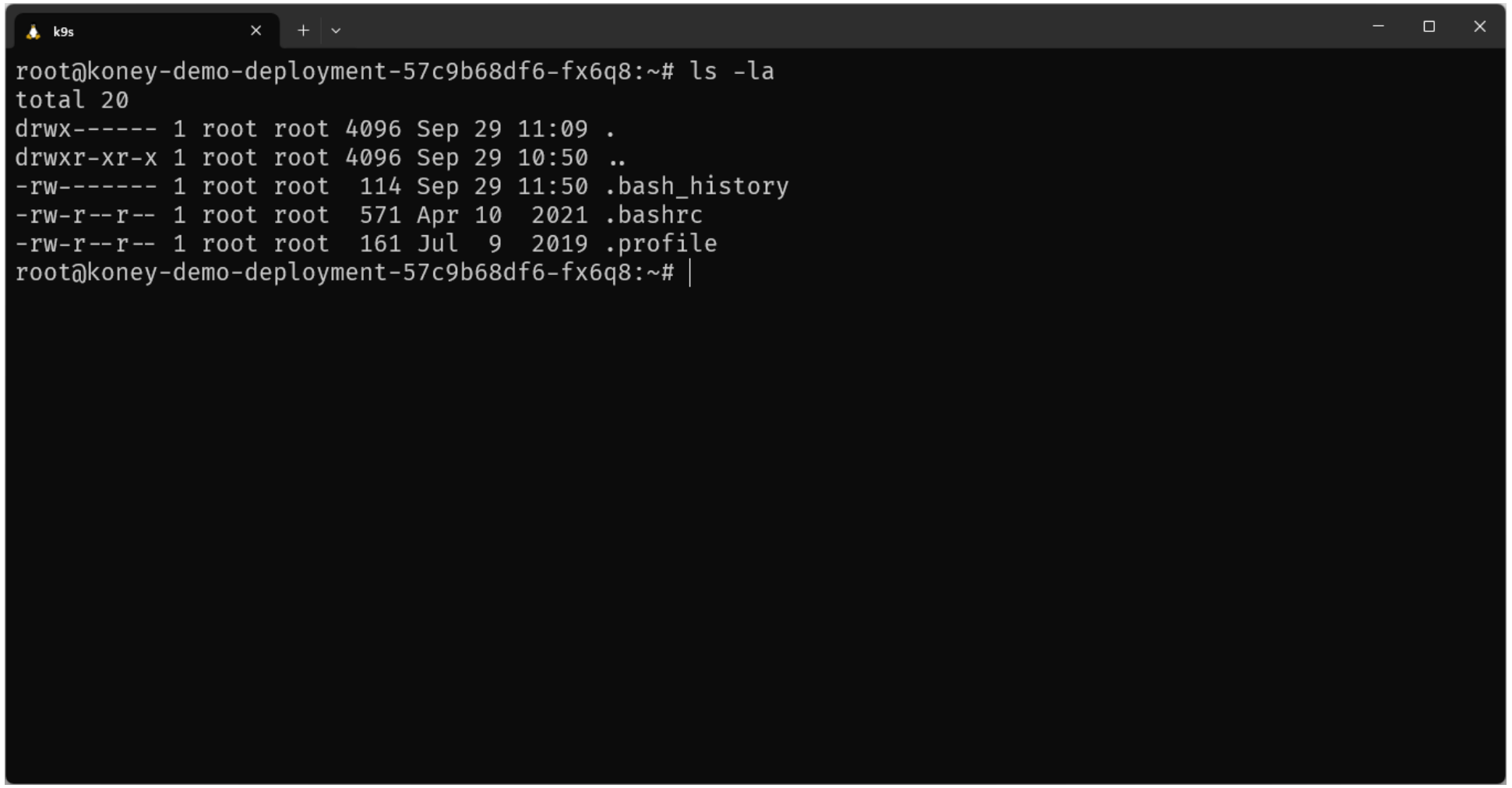


INTRODUCTION

You’re the attacker. Where should we go first?



~/

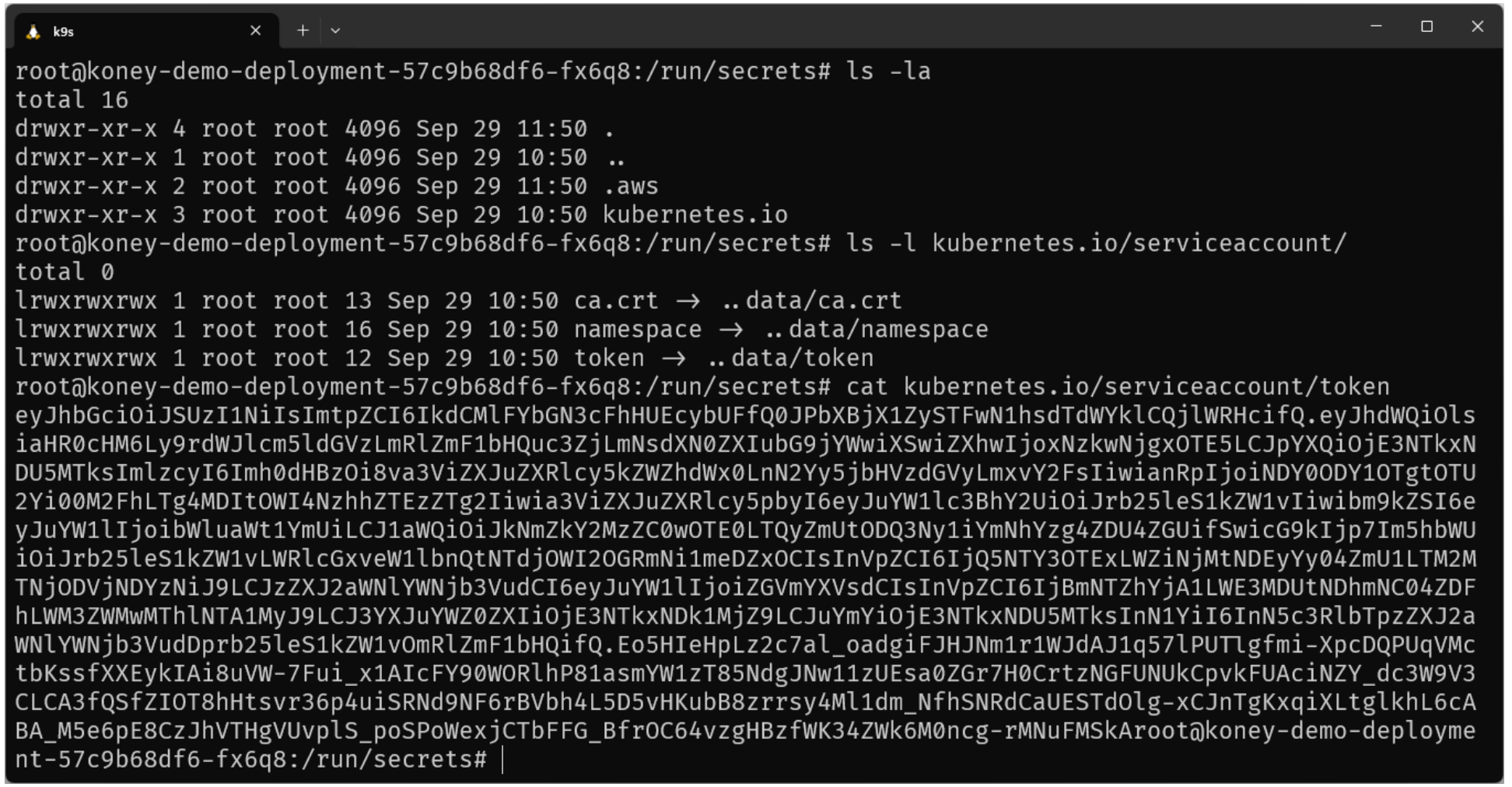

k9s
root@koney-demo-deployment-57c9b68df6-fx6q8:~# ls -la
total 20
drwx------ 1 root root 4096 Sep 29 11:09 .
drwxr-xr-x 1 root root 4096 Sep 29 10:50 ..
-rw------- 1 root root 114 Sep 29 11:50 .bash_history
-rw-r--r-- 1 root root 571 Apr 10 2021 .bashrc
-rw-r--r-- 1 root root 161 Jul 9 2019 .profile
root@koney-demo-deployment-57c9b68df6-fx6q8:~#

/run/secrets/

```
k9s
root@koney-demo-deployment-57c9b68df6-fx6q8:/run/secrets# ls -la
total 16
drwxr-xr-x 4 root root 4096 Sep 29 11:50 .
drwxr-xr-x 1 root root 4096 Sep 29 10:50 ..
drwxr-xr-x 2 root root 4096 Sep 29 11:50 .aws
drwxr-xr-x 3 root root 4096 Sep 29 10:50 kubernetes.io
root@koney-demo-deployment-57c9b68df6-fx6q8:/run/secrets#
```

If an attacker breaks into your container, they may search for sensitive information.

/run/secrets/kubernetes.io/serviceaccount/



If an attacker breaks into your container, they may search for sensitive information.

/run/secrets/.aws/

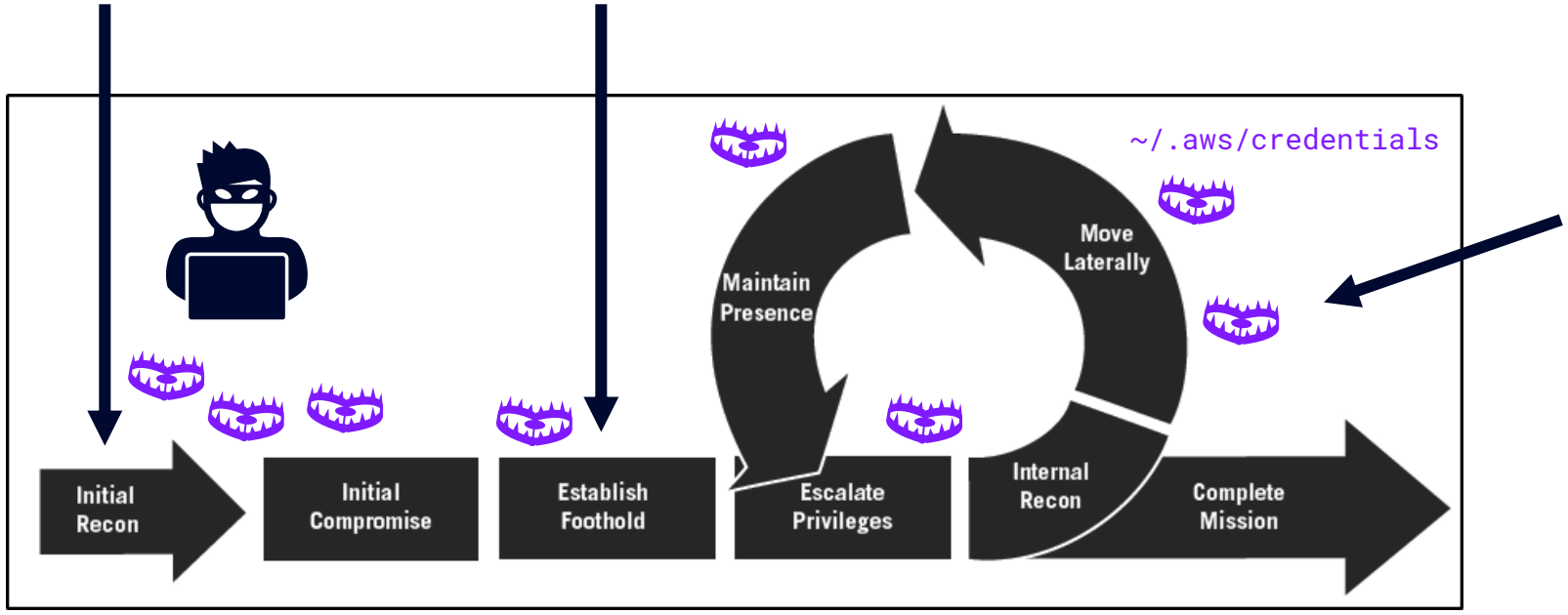
```
k9s
root@koney-demo-deployment-57c9b68df6-fx6q8:/run/secrets# ls -la
total 16
drwxr-xr-x 4 root root 4096 Sep 29 11:50 .
drwxr-xr-x 1 root root 4096 Sep 29 10:50 ..
drwxr-xr-x 2 root root 4096 Sep 29 11:50 .aws
drwxr-xr-x 3 root root 4096 Sep 29 10:50 kubernetes.io
root@koney-demo-deployment-57c9b68df6-fx6q8:/run/secrets# ls -l .aws
total 4
-r--r--r-- 1 root root 371 Sep 29 11:50 credentials
root@koney-demo-deployment-57c9b68df6-fx6q8:/run/secrets# cat .aws/credentials
[default]
aws_access_key_id = FZWGQYZRVMVQKCWXWJ
aws_secret_access_key = ToDNS2V5wkEuPmfocmNDHiVBspaywo
region = us-east-1

[staging]
aws_access_key_id = BPDHLBUJTYMKRVBMNB
aws_secret_access_key = Jf4mTvTG5zbLksf464nYwEVyDkbjdp
region = eu-central-1

[prod]
aws_access_key_id = GQRMYWAXADTVUPFMZJ
aws_secret_access_key = gUFQzyJWJSkMkWpay4Umxzg4oFkkju
region = us-east-1
root@koney-demo-deployment-57c9b68df6-fx6q8:/run/secrets#
```

If an attacker breaks into your container, they may search for sensitive information.

INTRODUCTION
What is Cyber Deception?

Attacks start with reconnaissance

Sophisticated adversaries will eventually find a way to infiltrate your network



[1]

We can place honeytokens, install fake endpoints, and inject deceptive content to …

- slow-down attackers by expending their “energy”
- assist defenders with strong indicators of compromise

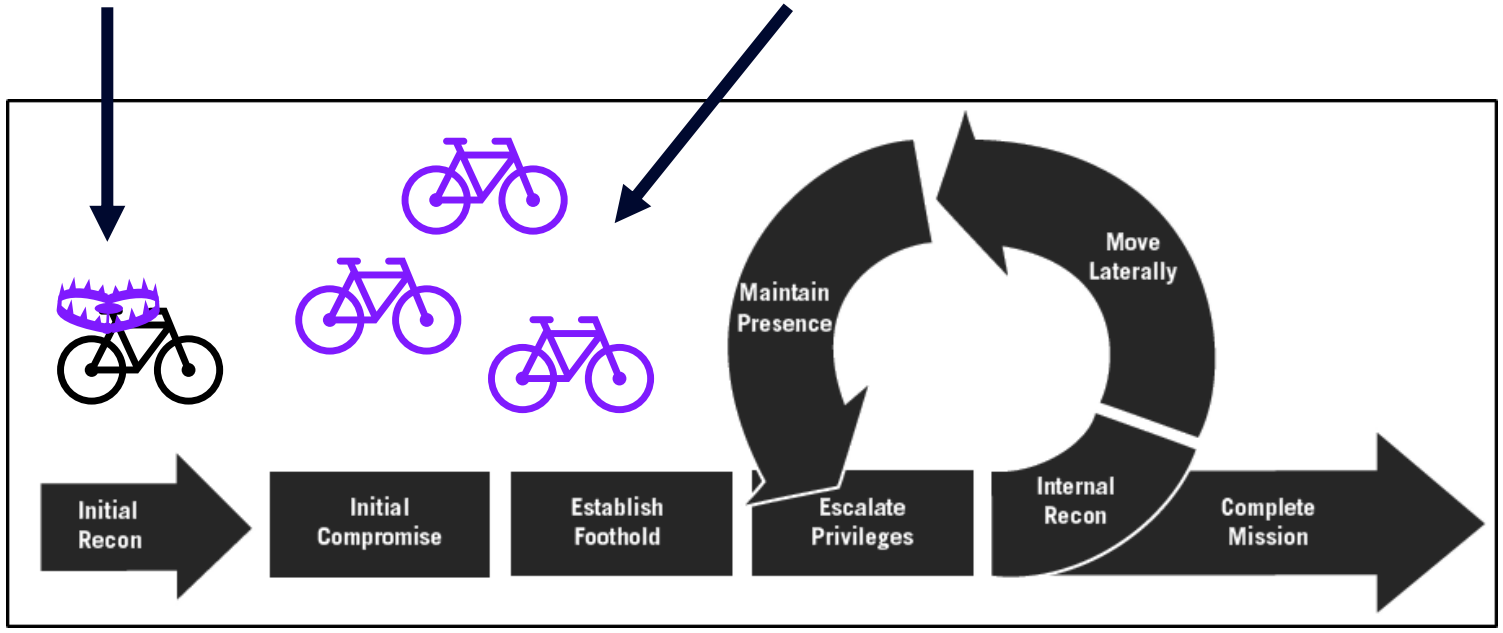
[1] Mandiant, “APT1: Exposing One of China’s Cyber Espionage Units,” 2013. [Online].

INTRODUCTION

Honeypots and Honeytokens

Thieves shall not steal your bike

Let's place some fake bikes then



[1]

'Classic' honeypots are isolated fake applications, reachable over the network (e.g., to collect threat intel)

Talk Focus

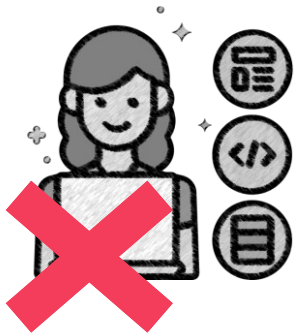
Honeytokens & application layer deception techniques are traps embedded into applications & systems

[1] Mandiant, "APT1: Exposing One of China's Cyber Espionage Units," 2013. [Online].

APPLICATION LAYER CYBER DECEPTION

Let's Embed Traps Directly Into Applications!

- Place **Honeytokens** in (container) filesystems
- Add new **HTTP endpoints** to mislead attackers
- Modify **HTTP headers**, e.g., version numbers
- Modify **HTTP bodies**, e.g., hidden form fields
- Other (non-HTTP) methods



But software applications are rarely deployed by the team that wrote the code, and often the responsibility for security measures lies entirely elsewhere.

Traps in Web Applications

- **Fixed HTTP Responses**
  - URL Redirection
  - Static Page, Honeyurl
    - Upload Sinkhole
- **HTTP Header Modification**
  - Version Trickery, Fake Banner
  - Status Code Tampering
  - Cookie Tampering
- **HTTP Body Modification**
  - robots.txt Modification, Disallow Injection
  - HTML, CSS, JS Modification
    - Tracking Links
    - Code Obfuscation
    - Weak Code Injection
    - GET / POST Parameter Injection
      - Hidden Form Fields
- **Other Methods (Non-HTTP, Custom Apps, etc.)**

Traps in File Systems

- **Honeyfiles**
  - Honeytokens
  - Honeydocuments
  - Honeydirectories

[2]

[2] M. Kahlhofer, M. Golinelli, and S. Rass, "Koney: A Cyber Deception Orchestration Framework for Kubernetes," in EuroS&PW '25. Venice, Italy: IEEE, 2025, pp. 690–702. doi: 10.1109/EuroSPW67616.2025.00084.

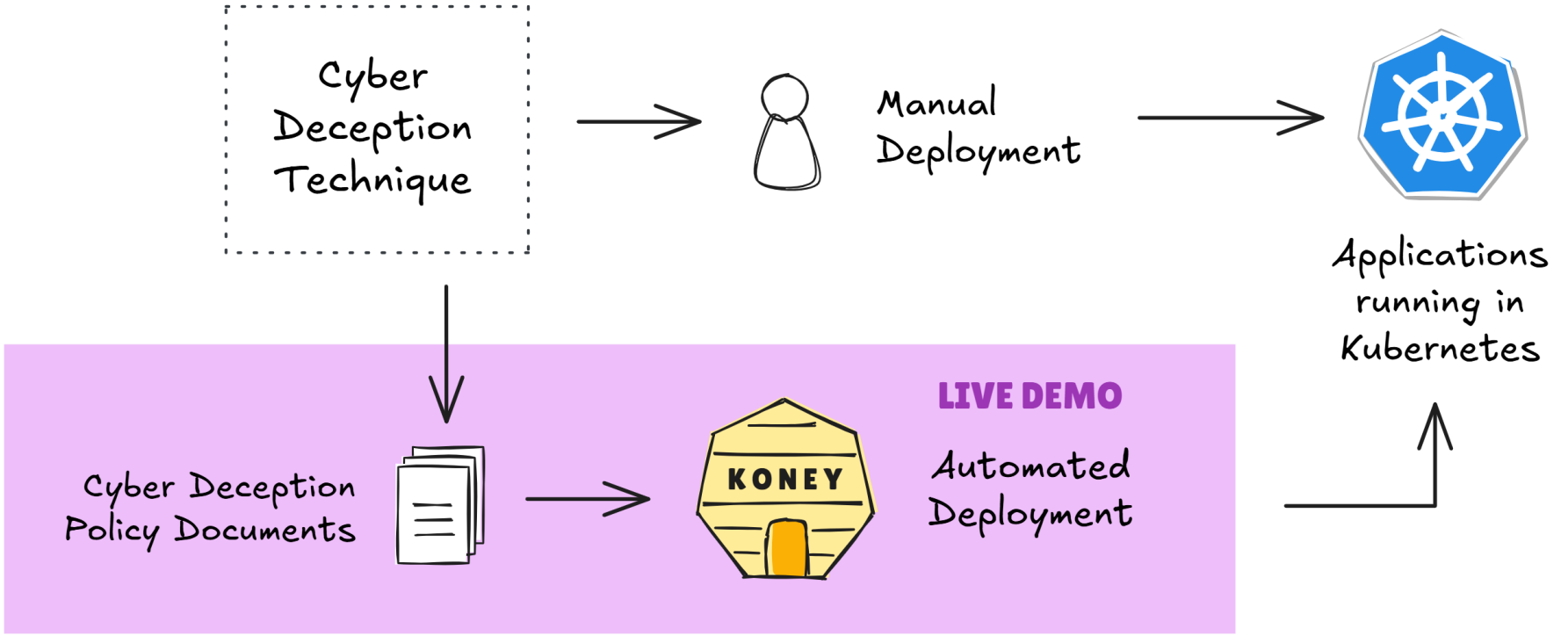
THE IDEA

“Deception-As-Code” with Koney [2]

github.com/dynatrace-oss/koney

GitHub

… instead of manually deciphering techniques from academic papers.



[2] M. Kahlhofer, M. Golinelli, and S. Rass, “Koney: A Cyber Deception Orchestration Framework for Kubernetes,” in EuroS&PW ’25. Venice, Italy: IEEE, 2025, pp. 690–702. doi: 10.1109/EuroSPW67616.2025.00084.

THE IDEA

Cyber Deception Policy Documents

github.com/dynatrace-oss/koney

GitHub

honeytoken.yaml — YAML

```yaml
apiVersion: koney.io/v1alpha1
kind: DeceptionPolicy
spec:
  traps:
    - filesystemHoneytoken:
        filePath: /run/secrets/token
        fileContent: "secret"
      match:
        any:
          - resources:
              selector:
                matchLabels:
                  op/honeytoken: true
  decoyDeployment:
    strategy: containerExec
  captorDeployment:
    strategy: tetragon
```

← Trap-specific parameterization

← Criteria for selecting the workloads (e.g., containers) in which to deploy the traps

← **Decoy. [3]** Strategy to deploy the trap itself

← **Captor. [3]** Strategy for monitoring the trap

[3] W. Fan, Z. Du, D. Fernández, and V. A. Villagrá, "Enabling an Anatomic View to Investigate Honeypot Systems: A Survey," IEEE Sys. J., vol. 12, no. 4, pp. 3906–3919, 2018, doi: 10.1109/jsyst.2017.2762161.

LIVE DEMO

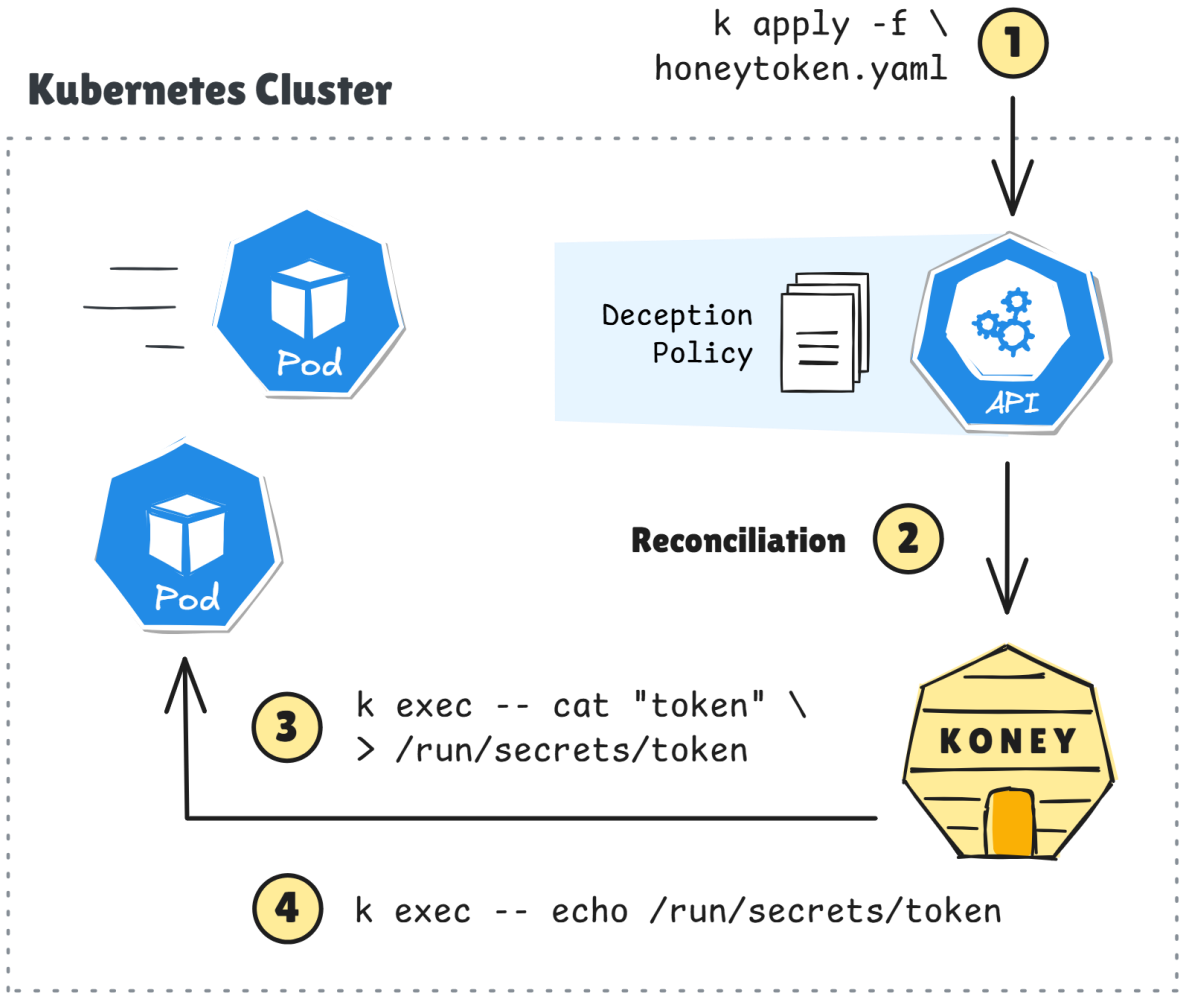
DeceptionPolicy

LIVE DEMO

More Traps!

DECOY STRATEGY

Placing Honeytokens by Executing Shell Commands



honeytoken.yaml — YAML

```yaml
apiVersion: koney.io/v1alpha1
kind: DeceptionPolicy
spec:
  traps:
    - filesystemHoneytoken:
        filePath: /run/secrets/token
        fileContent: "secret"
      match:
        any:
          - resources:
              selector:
                matchLabels:
                  op/honeytoken: true
      decoyDeployment:
        strategy: containerExec
      captorDeployment:
        strategy: tetragon
```

Placing Honeytokens by Executing Shell Commands (cont.)

Deployment	`cat "secret" > /run/secrets/token`
Verification	`echo /run/secrets/token`
Clean-Up	`rm /run/secrets/token`

Monitoring of access attempts	?
Transparency for system operators	✓ **DeceptionPolicy**
Zero Downtime of application services	✓ yes
Non-interference with genuine operation	✓ just a few process executions

LIVE DEMO

File Access Monitoring

LIVE DEMO

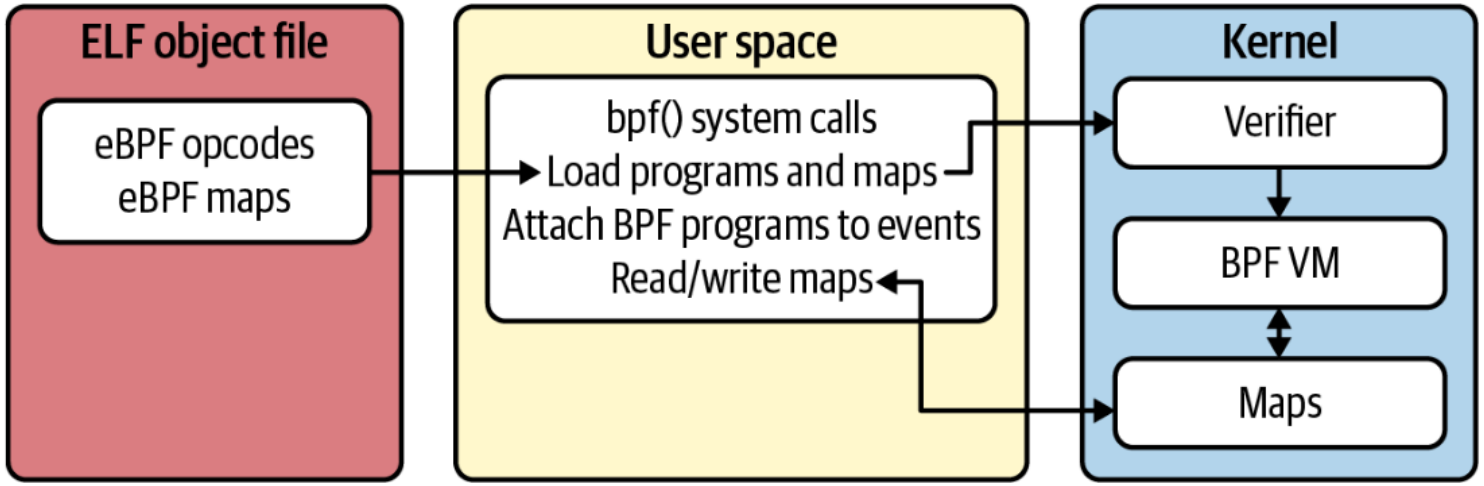
Koney Alert Example

alert.json JSON


```
{
  "timestamp": "2025-06-02T11:17:02Z",
  "deception_policy_name": "deceptionpolicy-aws-credentials",
  "trap_type": "filesystem_honeytoken",
  "metadata": { "file_path": "/run/secrets/.aws/credentials" },
  "pod": {
    "name": "koney-demo-deployment-5bcbd78875-45qpn",
    "namespace": "koney-demo",
    "container": {
      "id": "docker://e19c1827e255ce7a5c5fd74eb4ee861388f83a16410effd65e30d3b051cd815f",
      "name": "nginx"
    }
  },
  "process": {
    "pid": 148373, "uid": 0, "cwd": "/", "binary": "/usr/bin/cat",
    "arguments": "/run/secrets/.aws/credentials"
  }
}
```

File Access Monitoring with eBPF

eBPF makes the kernel programmable. eBPF programs are typically written in a subset of C or Rust and compiled to an object file.

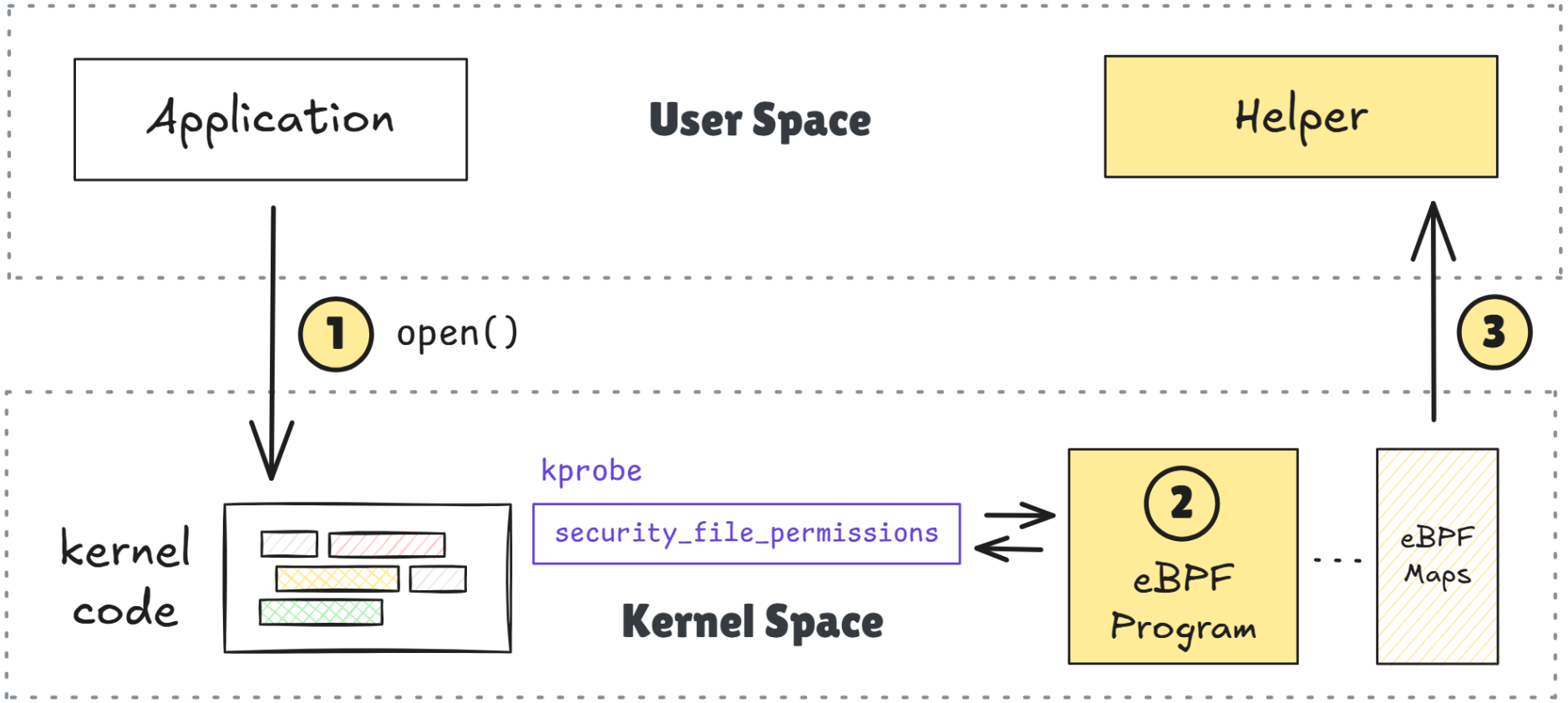


[4]

[4] L. Rice, What Is eBPF? Sebastopol, CA: O'Reilly Media, Inc., 2022. Available: https://isovalent.com/books/ebpf/

File Access Monitoring with eBPF (cont.)

We hook the **security_file_permissions** kprobe in kernel space.



File Access Monitoring with eBPF (cont.)

policy.yaml YAML

```yaml
apiVersion: cilium.io/v1alpha1
kind: TracingPolicy
metadata:
  name: monitor-honeytoken
spec:
  kprobes:
    - call: security_file_permission
      syscall: false
      return: true
      args:
        - index: 0
          type: file
        - index: 1
          type: int
      returnArg:
        index: 0
        type: int
      returnArgAction: Post
      selectors:
        - matchArgs:
            - index: 0
              operator: Prefix
              values:
                - /run/secrets/token
```

[5]

Tetragon is a Kubernetes Operator that simplifies the creation of "tracing policies" in K8s clusters.

Falco and **Tracee** are popular alternatives.

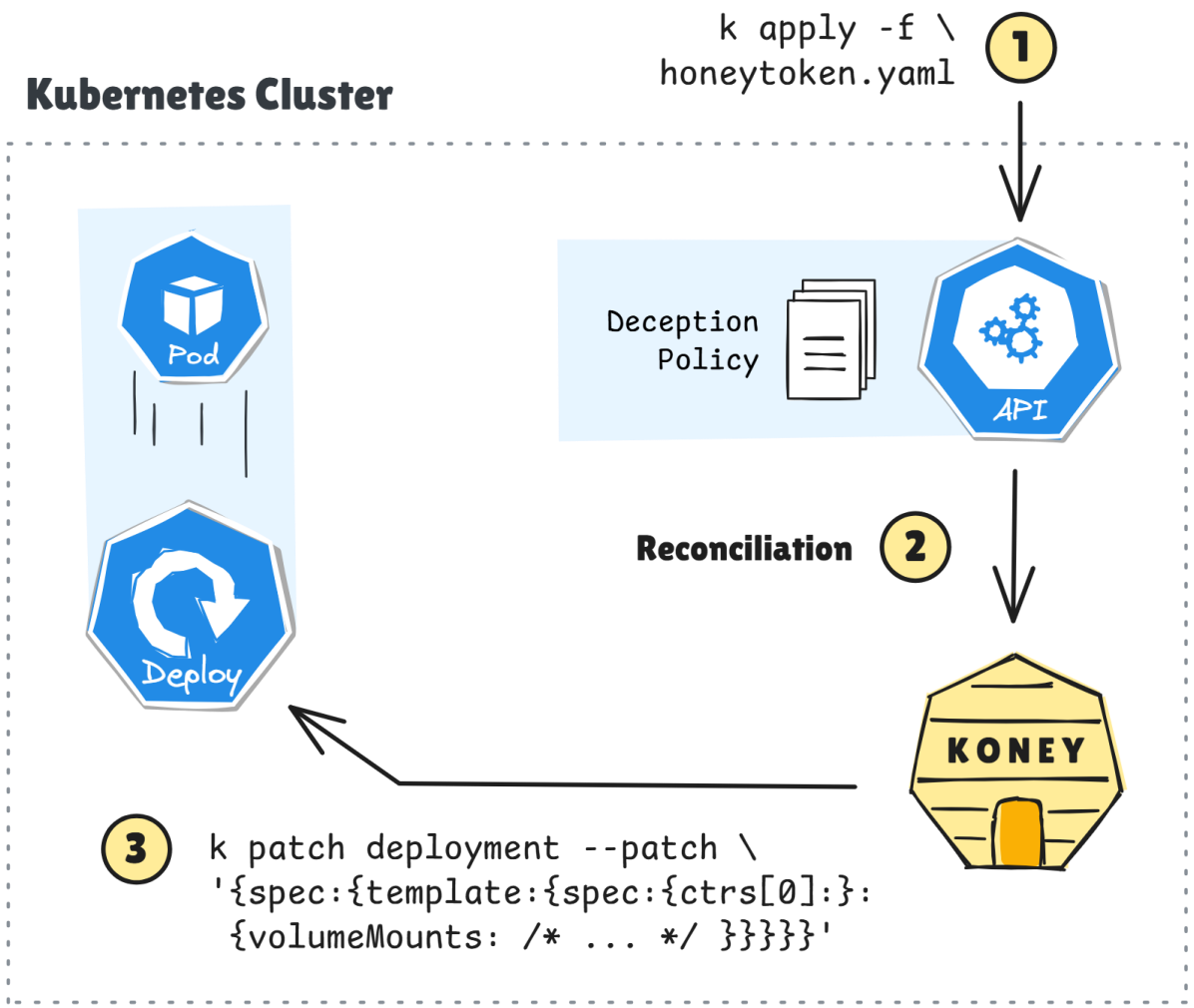
[5] K. Kourtis and A. Papagiannis, "File Monitoring with eBPF and Tetragon (Part 1)," Isovalent Blog. Accessed: Feb. 2025. Available: https://isovalent.com/blog/post/file-monitoring-with-ebpf-and-tetragon-part-1/

LIVE DEMO

Distroless Images

DECOY STRATEGY

Placing Honeytokens by Mounting Volumes



deployment.yaml — YAML

```yaml
apiVersion: apps/v1
kind: Deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - image: nginx:latest
          name: web
          volumeMounts:
            - mountPath: /run/secrets/token
              name: koney-volume
              readOnly: true
      volumes:
        - name: koney-volume
          secret:
            secretName: koney-secret
```

Placing Honeytokens With Koney

	Strategy 1: Shell Commands	Strategy 2: Volume Mounts
Deployment	`cat "secret" > /run/secrets/token`	`+spec.containers.volumeMounts`
Verification	`echo /run/secrets/token`	`echo /run/secrets/token`
Clean-Up	`rm /run/secrets/token`	`-spec.containers.volumeMounts`
Monitoring of access attempts	✓ **eBPF (via Tetragon)**	✓ **eBPF (via Tetragon)**
Transparency for system operators	✓ **DeceptionPolicy**	✓ even better, visible manifest change
Zero Downtime of application services	✓ yes	✗ **needs container restart in Kubernetes**
Non-interference with genuine operation	✓ just a few process executions	✓ even better, no process executions

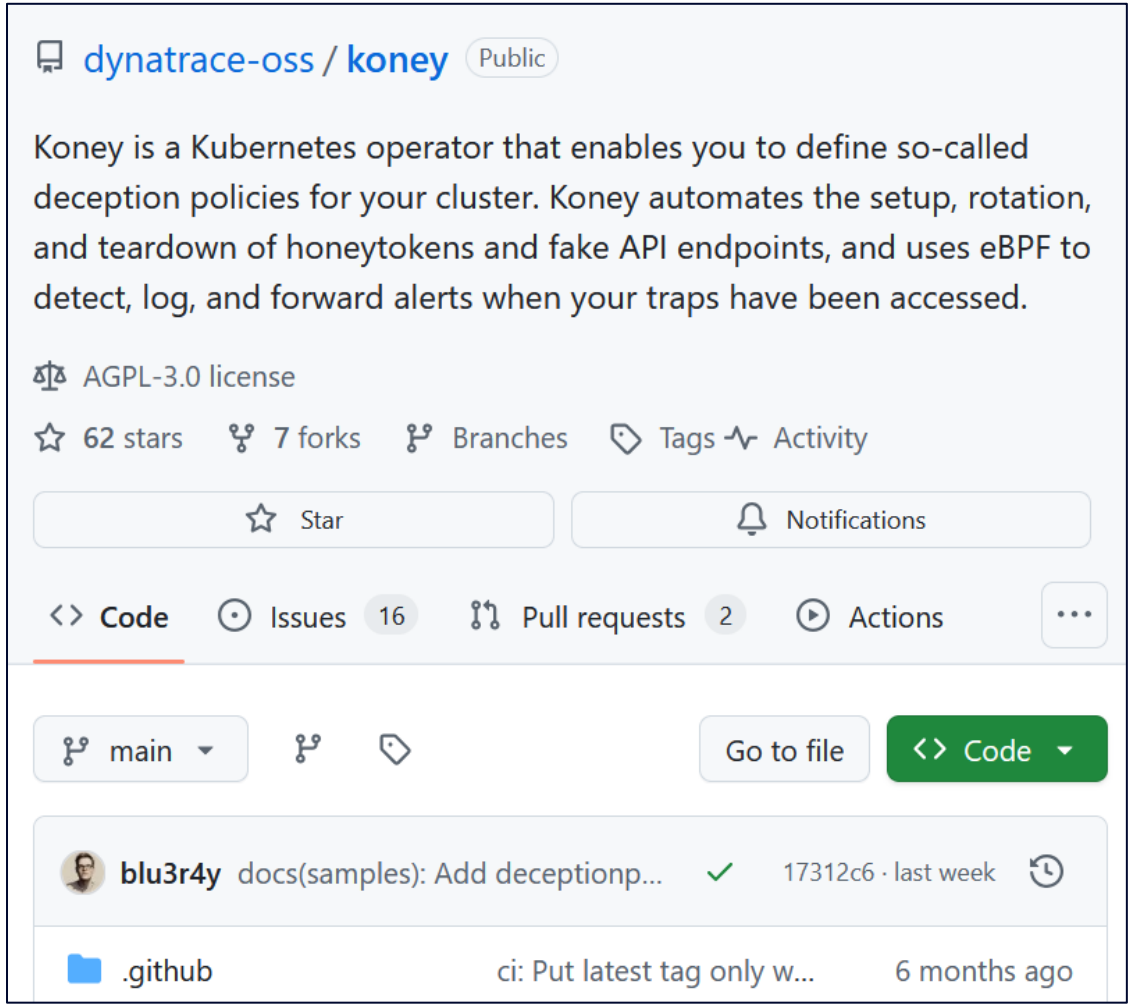
OUTLOOK

Deceive. Test. Repeat.

github.com/dynatrace-oss/koney

GitHub

Let's provide **app developers** and **system operators** with a refreshingly different concept on detecting attackers!



Development Outlook

- Traps for HTTP-based apps
- eBPF Monitoring w/o Tetragon

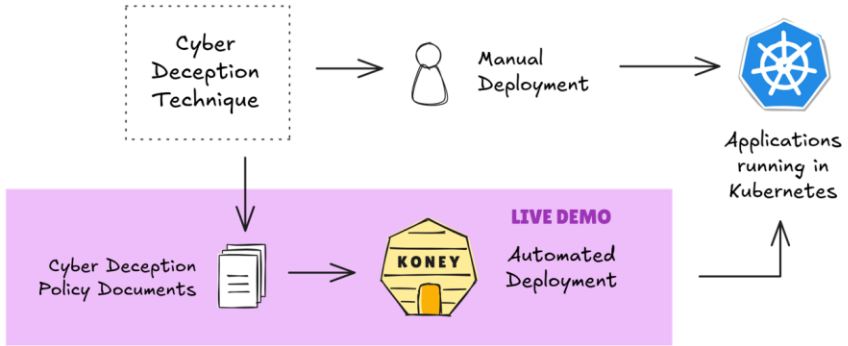
THE IDEA

"Deception-As-Code" with Koney [2]

github.com/dynatrace-oss/koney

GitHub

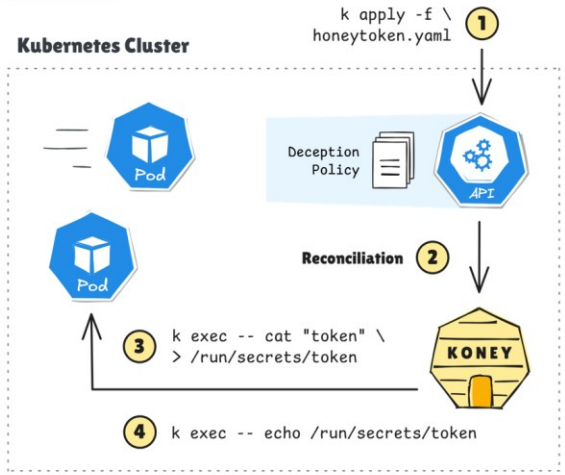
… instead of manually deciphering techniques from academic papers.



[2] M. Kahlhofer, M. Golinelli, and S. Rass, "Koney: A Cyber Deception Orchestration Framework for Kubernetes," in EuroS&PW '25. Venice, Italy: IEEE, 2025, pp. 690–702. doi: 10.1109/EuroSPW67616.2025.00084.

DECOY STRATEGY

Placing Honeytokens by Executing Shell Commands



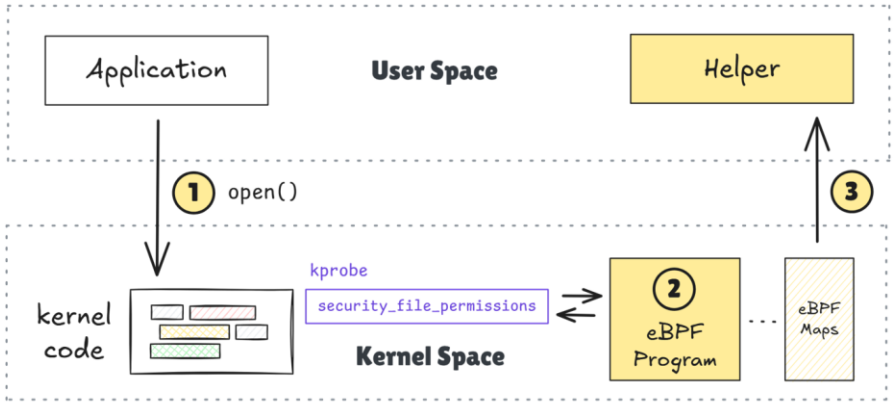
honeytoken.yaml

```yaml
apiVersion: koney.io/v1alpha1
kind: DeceptionPolicy
spec:
  traps:
    - filesystemHoneytoken:
        filePath: /run/secrets/token
        fileContent: "secret"
      match:
        any:
          - resources:
              selector:
                matchLabels:
                  op/honeytoken: true
      decoyDeployment:
        strategy: containerExec
      captorDeployment:
        strategy: tetragon
```


CAPTOR STRATEGY

File Access Monitoring with eBPF (cont.)

We hook the **security_file_permissions** kprobe in kernel space.



OUTLOOK

Deceive. Test. Repeat.

github.com/dynatrace-oss/koney

GitHub

Let's provide **app developers** and **system operators** with a refreshingly different concept on detecting attackers!

Development Outlook

- Traps for HTTP-based apps
- eBPF Monitoring w/o Tetragon