Beyond Honeypots

Practical Cyber Deception for Real Systems

IT Security Community Exchange

IT-SECX · October 3, 2025 · St. Pölten, AT

PRESENTER

Mario Kahlhofer

Senior Research Scientist
Dynatrace Research

IT SECX

dynatrace

JOHANNES KEPLER UNIVERSITÄT LINZ

whoami

Mario Kahlhofer

Senior Research Scientist · Dynatrace Research

PhD Student · Johannes Kepler University Linz

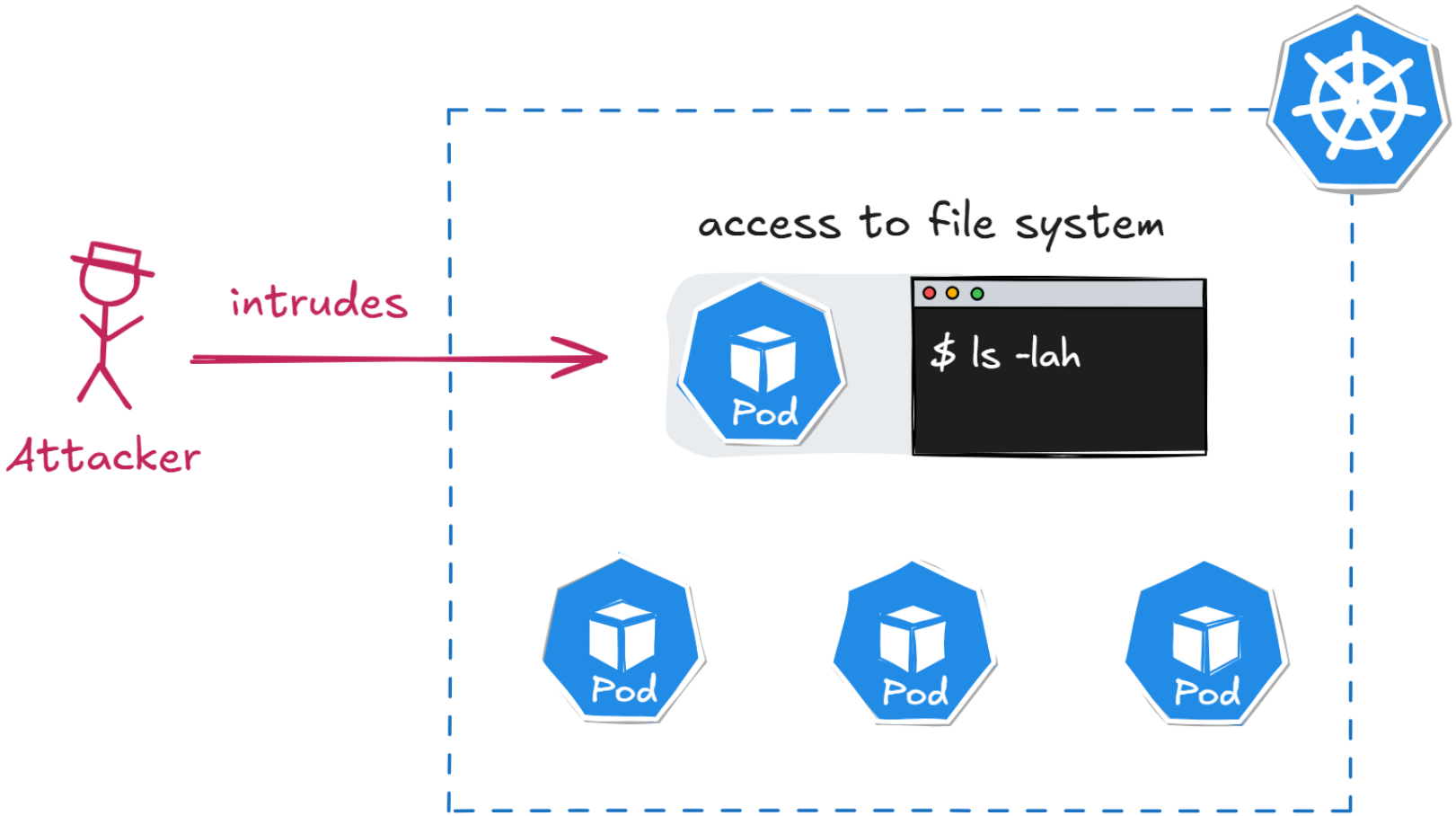
mario.kahlhofer@dynatrace.com

github.com/blu3r4y

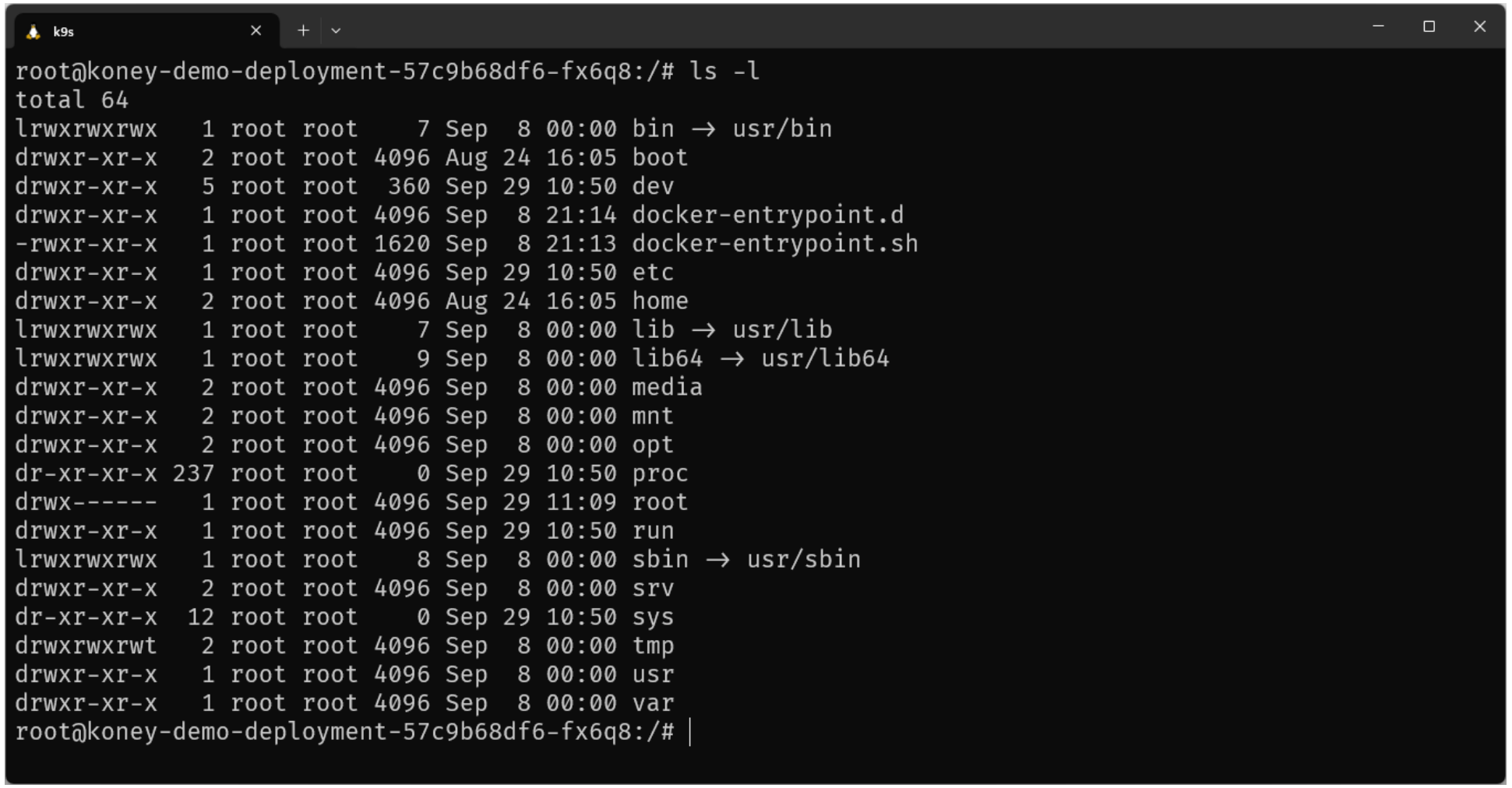
LinkedIn

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

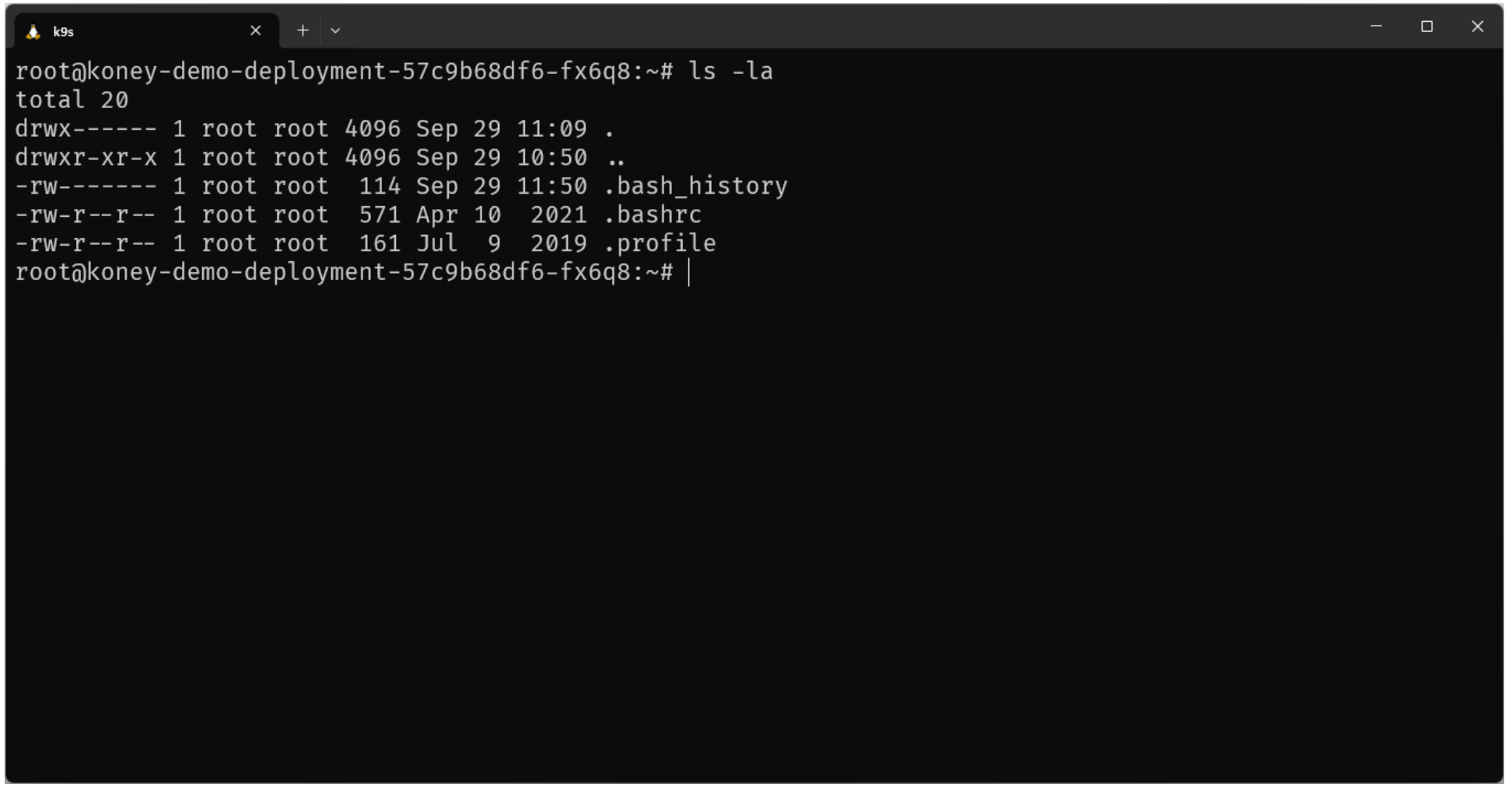
Imagine an attacker infiltrated your network …



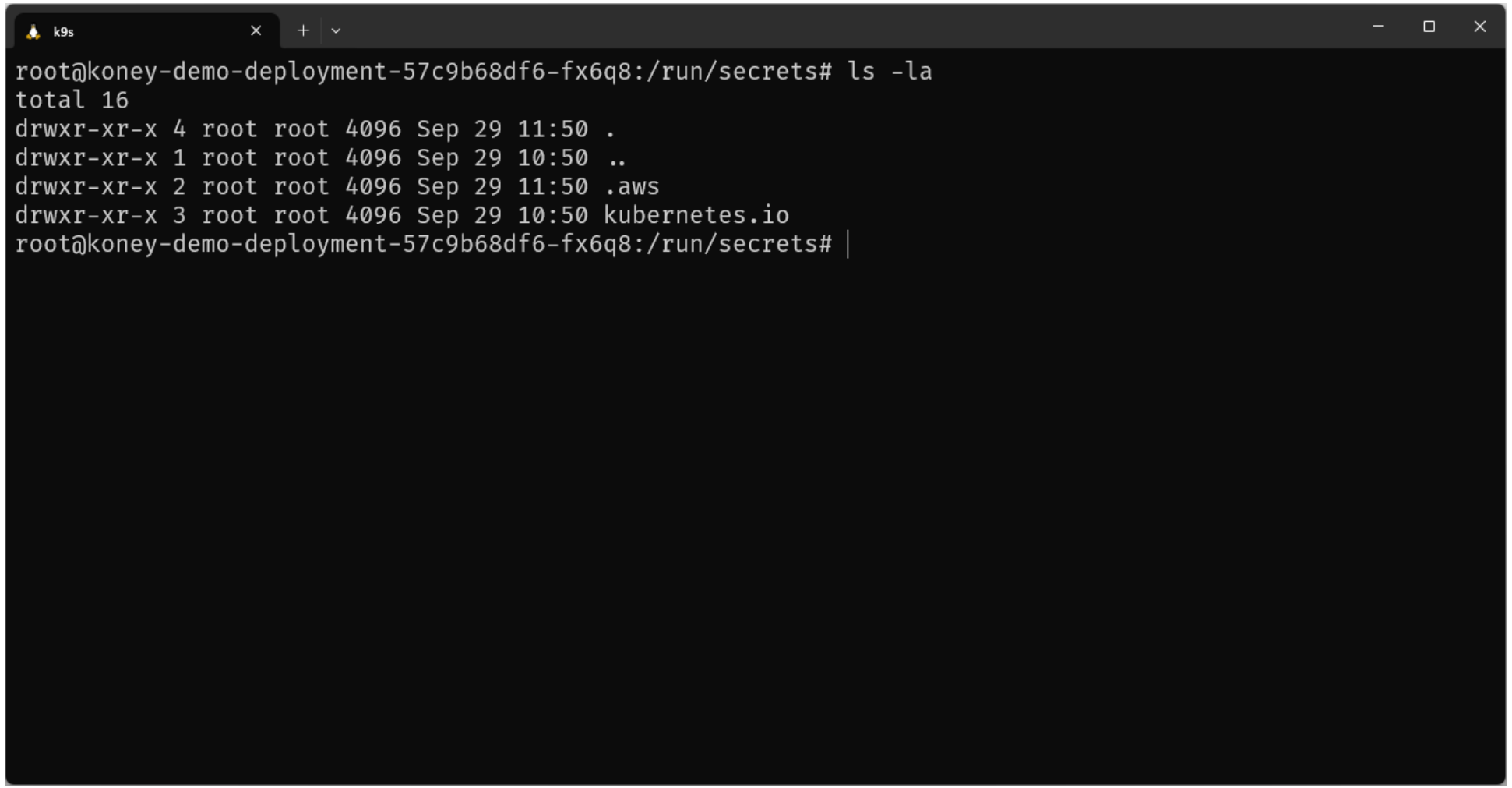
You’re the attacker. Where should we go first?



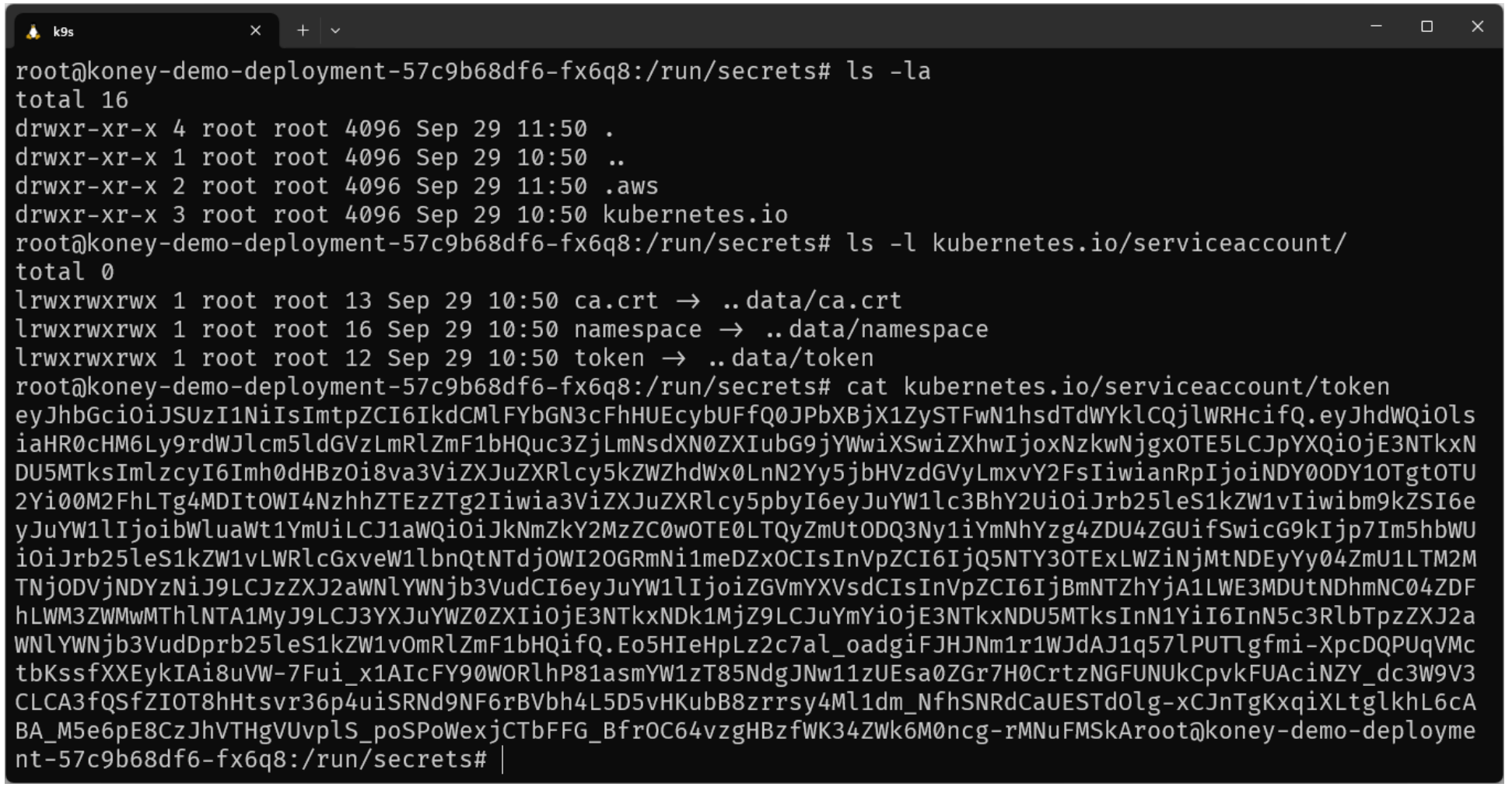
~/


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:~#

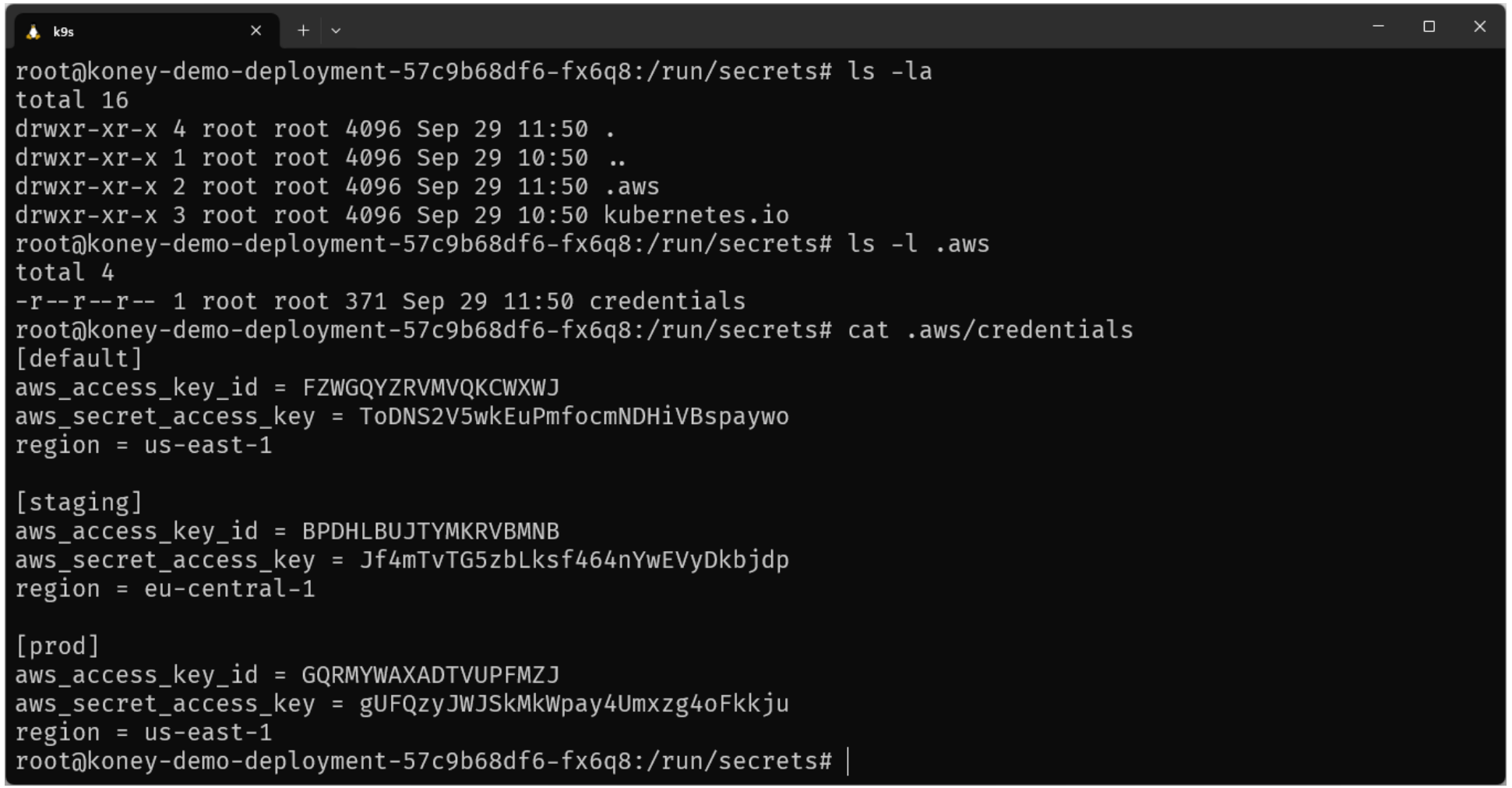
/run/secrets/


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#

/run/secrets/kubernetes.io/serviceaccount/



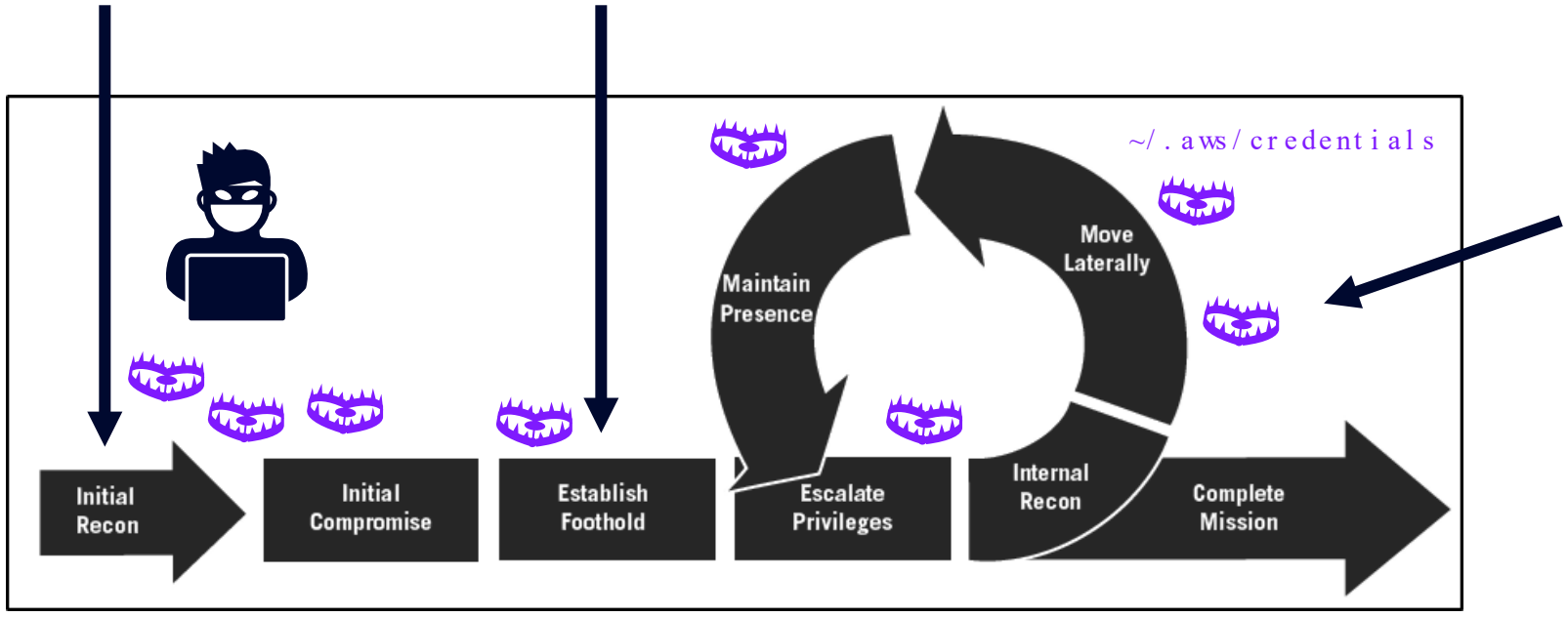
/run/secrets/.aws/



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 …

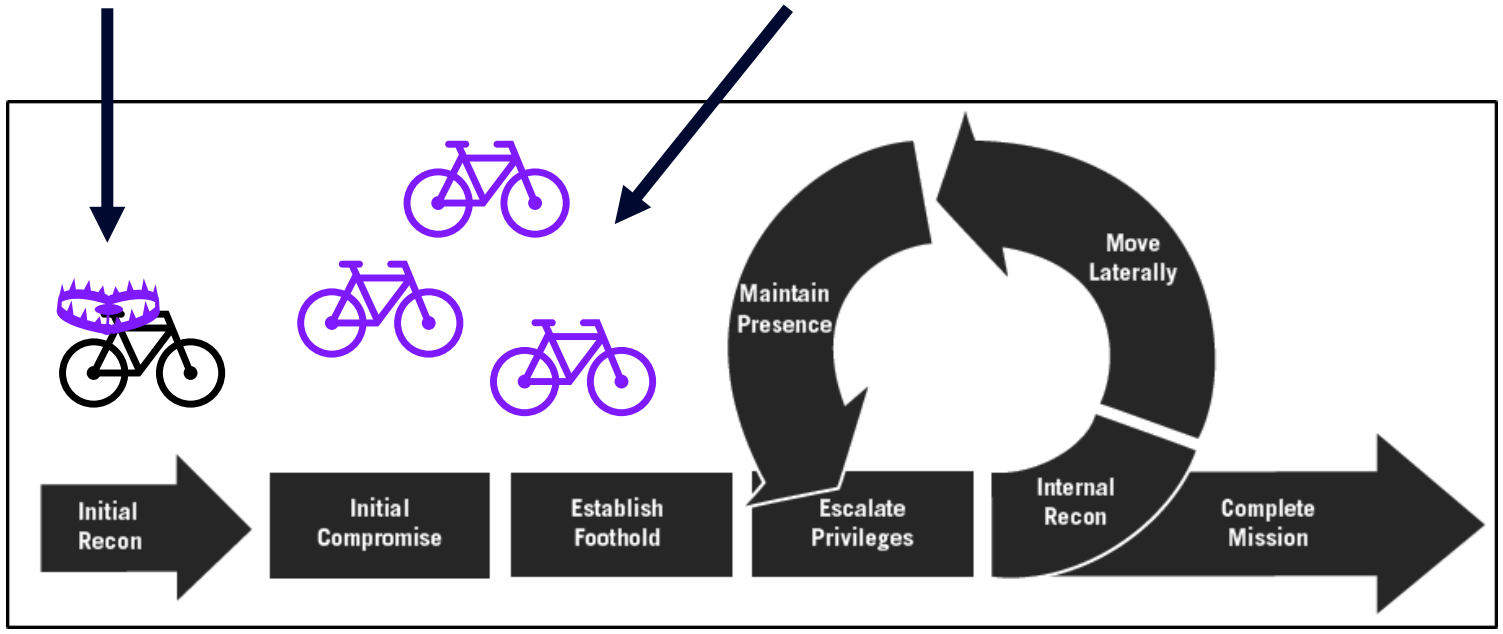
- **slow-down attackers** by expending their "energy"
- **assist defenders** with strong indicators of compromise

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

Honeypots and Honeytokens

Thieves shall not steal your bike

Let's place some fake bikes then



[1]

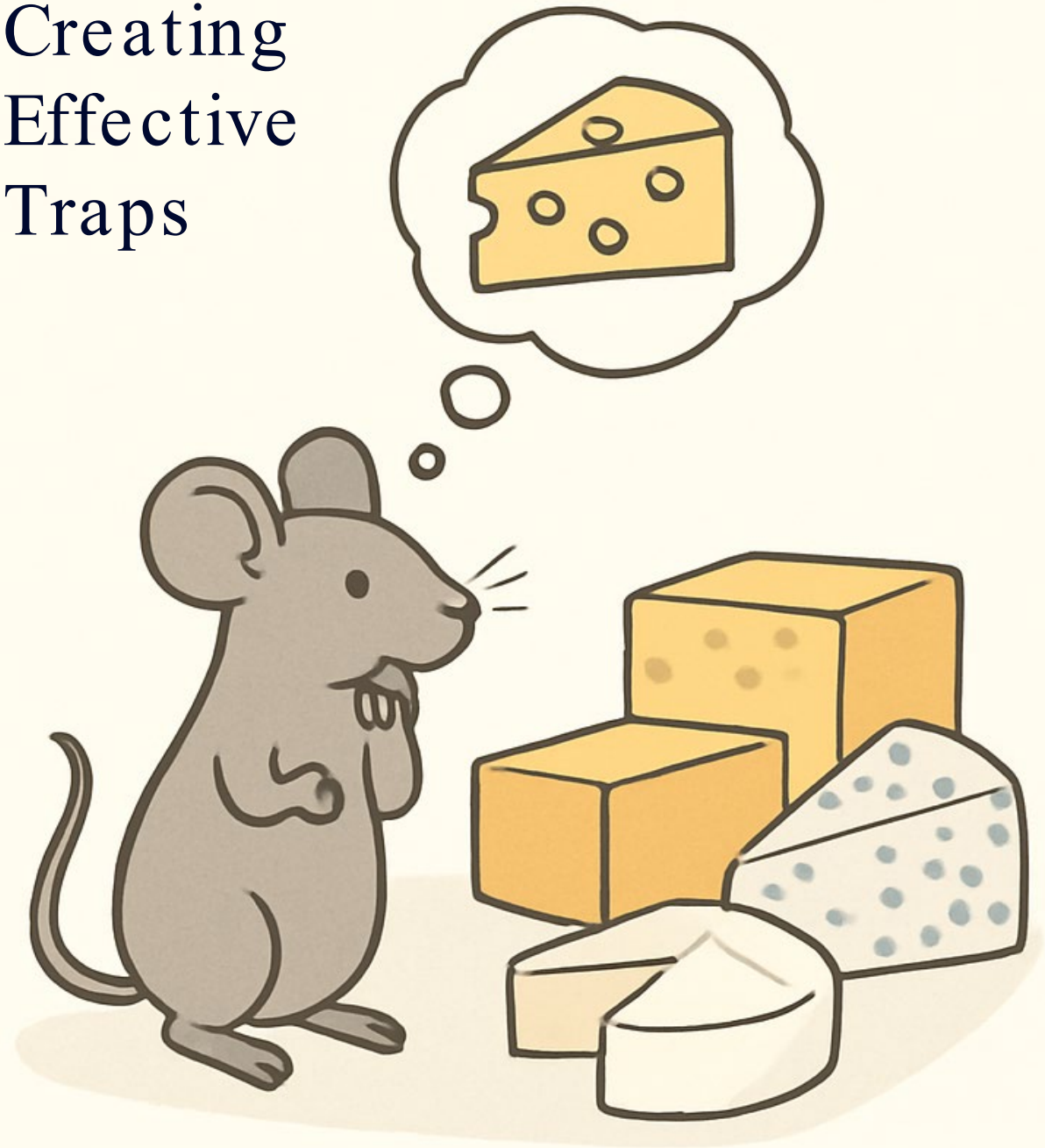
'Classic' honeypots are isolated fake applications, reachable over the network (e.g., to collect threat intel)

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].

Creating Effective Traps

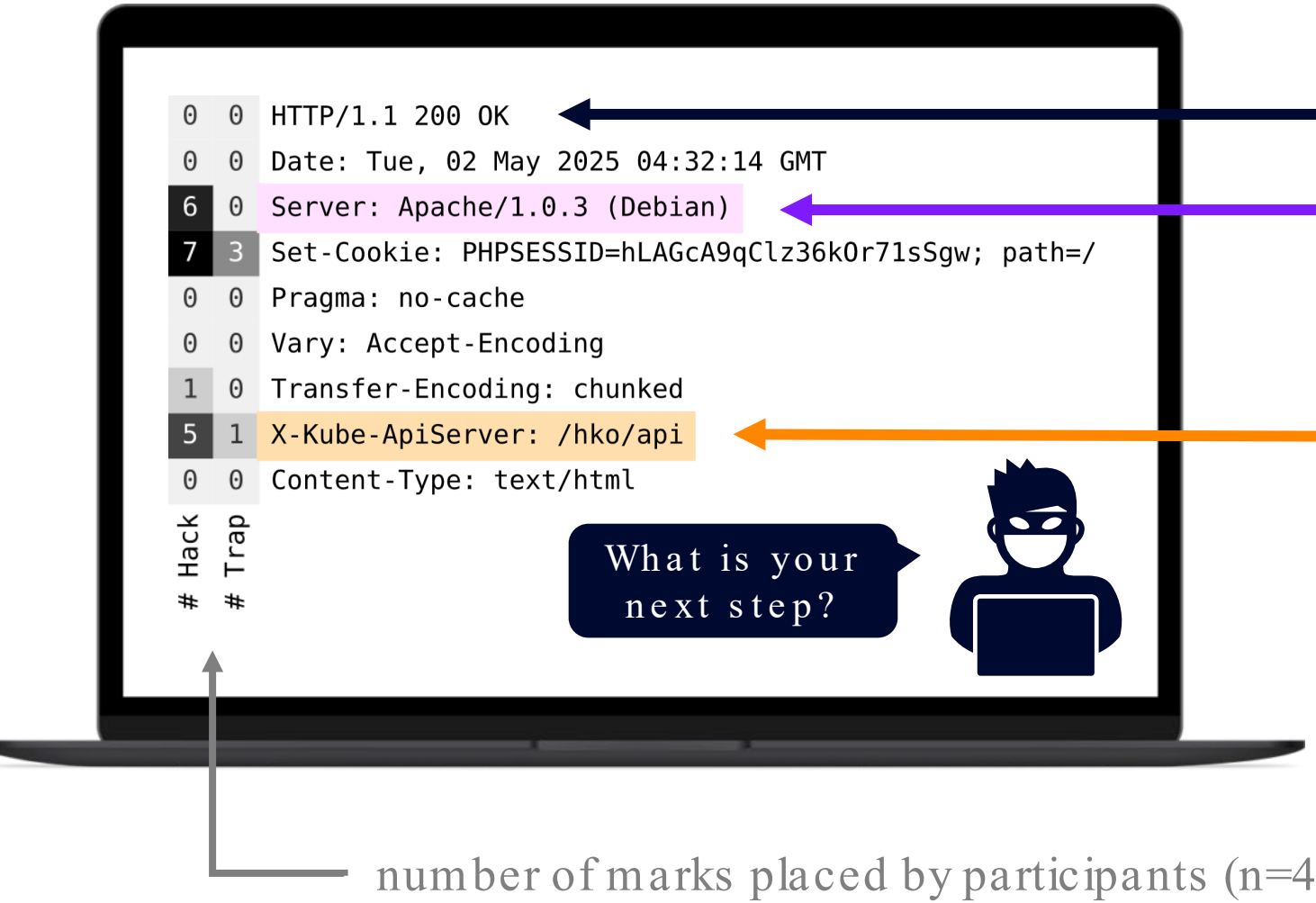


Deploying Traps

The Honeyquest Study [2]

github.com/dynatrace-oss/honeyquest

GitHub



Neutral Elements

Risky Element

e.g., a true indicator of vulnerability CVE-1999-0067

Deceptive Element

Traps injected by us

number of marks placed by participants (n=47)

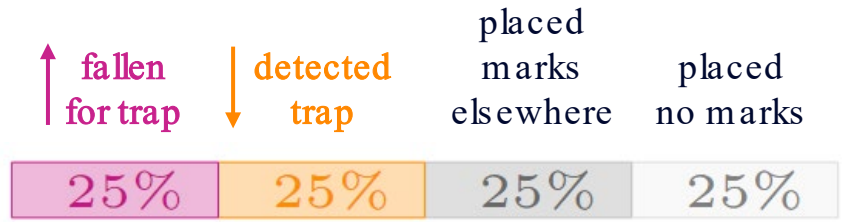
[2] M. Kahlhofer, S. Achleitner, S. Rass, and R. Mayrhofer, "Honeyquest: Rapidly Measuring the Enticingness of Cyber Deception Techniques with Code-based Questionnaires," in RAID '24. Padua, Italy: ACM, 2024, pp. 317–336. doi: 10.1145/3678890.3678897.

Which trap is better?

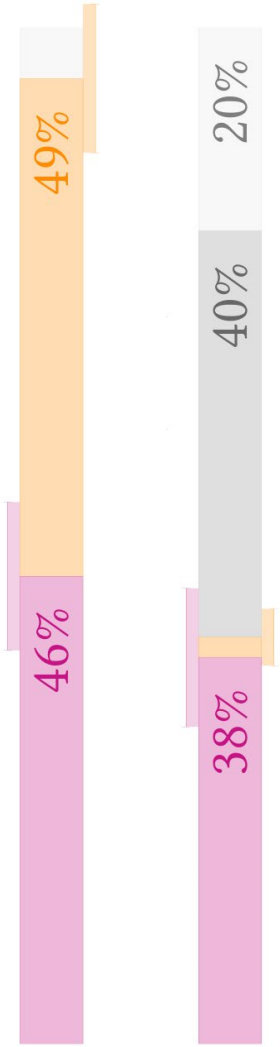
passwords.txt

config.ini

Which trap is better?



passwords.txt



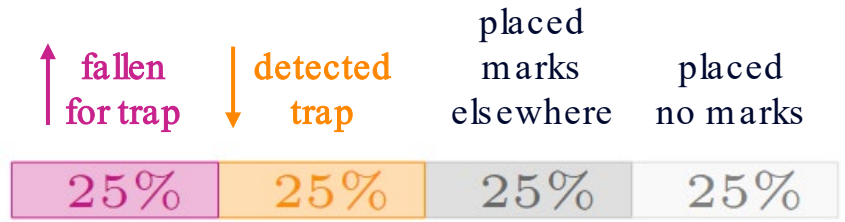
config.ini

Which trap is better?

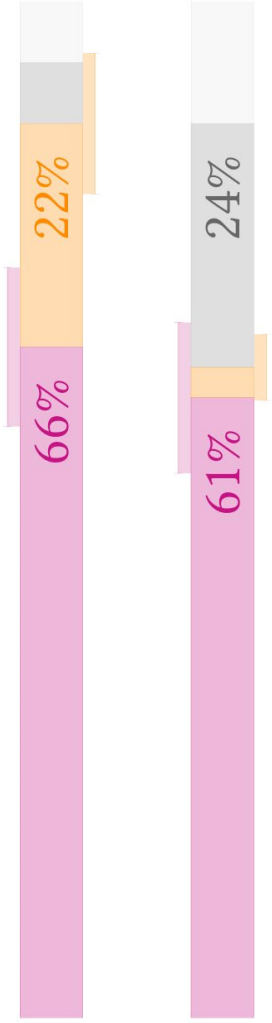
private-key.pem

backup.tar.gz

Which trap is better?



private-key.pem



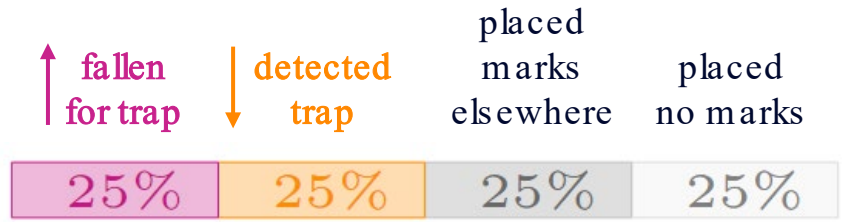
backup.tar.gz

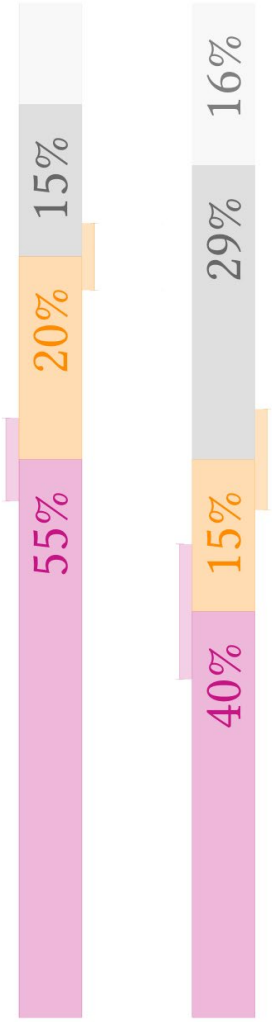
Which trap is better?

X-DevToken:
eyJhbGciIU.zI1NiI..

Set-Cookie:
DATA=YWRtaW49Zm2U%3D

Which trap is better?





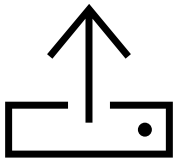
</>

X-DevToken:
eyJhbGciIU.zI1NiI..

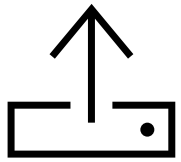
</>

Set-Cookie:
DATA=YWRtaW49Zm2U%3D

Which trap is better?

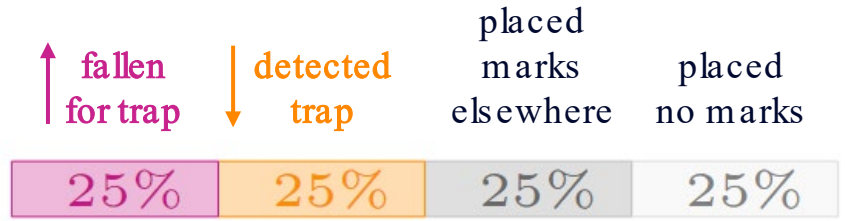


GET
https://github.com
/owner/my_repo
/settings/secrets
/58975/read
200 OK

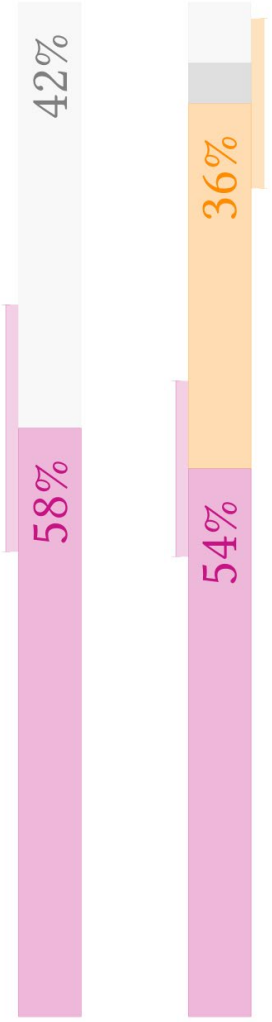


GET
https://github.com
/rest/user/login
?user=johnson
&password=carrot13
200 OK

Which trap is better?



GET
https://github.com
/owner/my_repo
/settings/secrets
/58975/read
200 OK



GET
https://github.com
/rest/user/login
?user=johnson
&password=carrot13
200 OK

The Honeyquest Study [2]

github.com/dynatrace-oss/honeyquest

Git Hub

File System	18	443	42% 21% 21% 16%	139	39%
DF1 private-key.pem	2	32	66% 22%	17	35%
DF2 backup.tar.gz	2	38	61% 24%	19	42%
DF3 keys.json	3	86	52% 26%	37	51%
DF4 card3rz_reg_details.html [75]	2	43	51% 19% 19%	15	47%
DF5 passwords.txt	2	41	46% 49%	12	25%
DF6 customer_list_2010.html [75]	2	53	43% 23% 25%	17	29%
DF7 config.ini	2	45	38% 40% 20%	15	33%
DF8 SPAM_list.pdf [75]	2	58	17% 21% 45% 17%	7	14%
DF9 Rowe et al. [83, 82]	1	47	15% 19% 57%	0	–
.htaccess Files	5	128	54% 21%	14	36%
DH1 Admin Redirect	5	128	54% 21%	14	36%
HTTP Responses	23	456	36% 30% 22%	103	24%
DP1 Developer Token	7	137	55% 20% 15%	46	33%
DP2 Cookie [54, 87, 88]	2	48	40% 15% 29% 16%	13	38%
DP3 API Server	7	134	30% 30% 30%	21	5%
DP4 Proxy Referer	7	137	22% 47% 24%	23	17%
HTTP Requests	25	534	30% 22% 37%	57	53%
DS1 IDOR Secrets [87]	1	12	58% 42%	2	–
DS2 Clear-Text Pass. [88]	1	28	54% 36%	12	67%
DS3 SESSID Param. [54]	2	58	40% 16% 26% 18%	10	70%
DS4 Path Traversal [88]	5	122	39% 20% 27%	10	40%
DS5 Admin Param. [54, 77]	2	49	37% 22% 27%	6	50%
DS6 Unescaped JS [88]	2	21	33% 57%	0	–
DS7 System Param. [54, 88]	1	28	29% 25% 42%	4	–
DS8 Developer Endpoint	4	80	24% 26% 42%	6	33%
DS9 Unespaced JSON [88]	2	22	23% 32% 40%	3	–
DS10 Mass Assignment	2	44	30% 54%	2	–
DS11 Log Endpoint	3	70	31% 59%	2	–

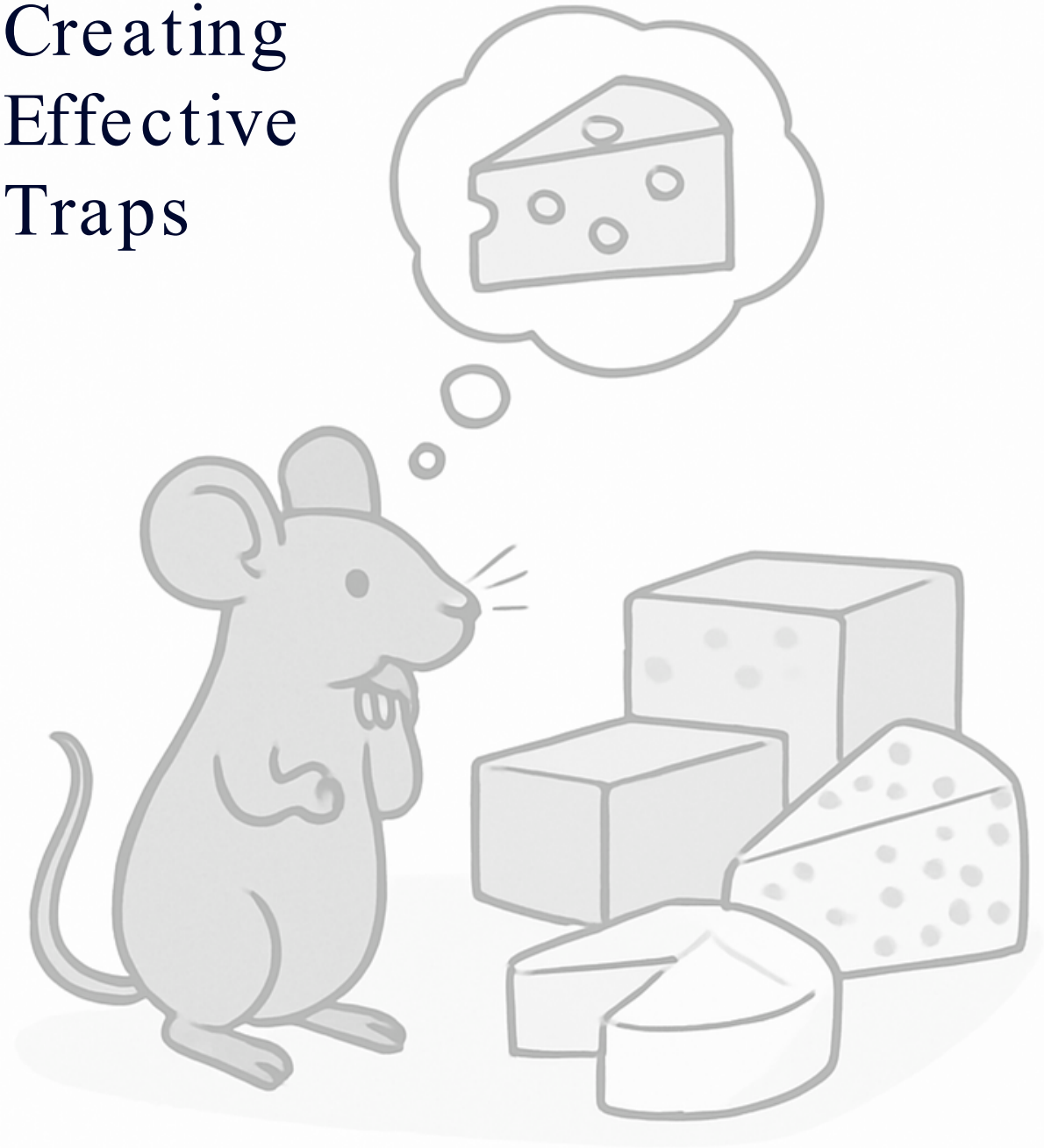
BEFORE AFTER

More research results and suggestions for designing "enticing" traps.

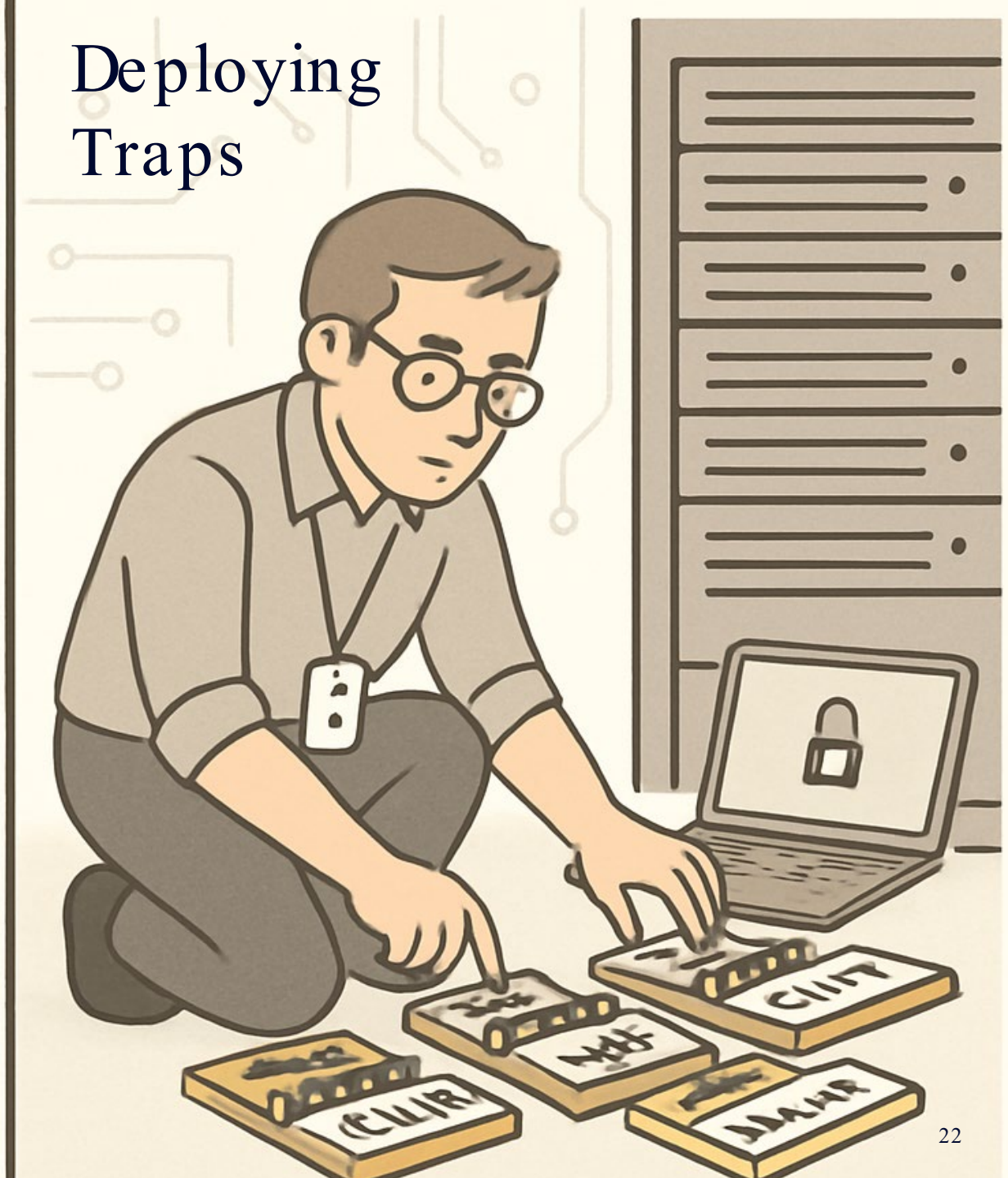
Research findings on placing traps that reduce the risk of exploiting genuine security weaknesses.

[2] M. Kahlhofer, S. Achleitner, S. Rass, and R. Mayrhofer, "Honeyquest: Rapidly Measuring the Enticingness of Cyber Deception Techniques with Code-based Questionnaires," in RAID '24. Padua, Italy: ACM, 2024, pp. 317–336. doi: 10.1145/3678890.3678897.

Creating Effective Traps



Deploying Traps

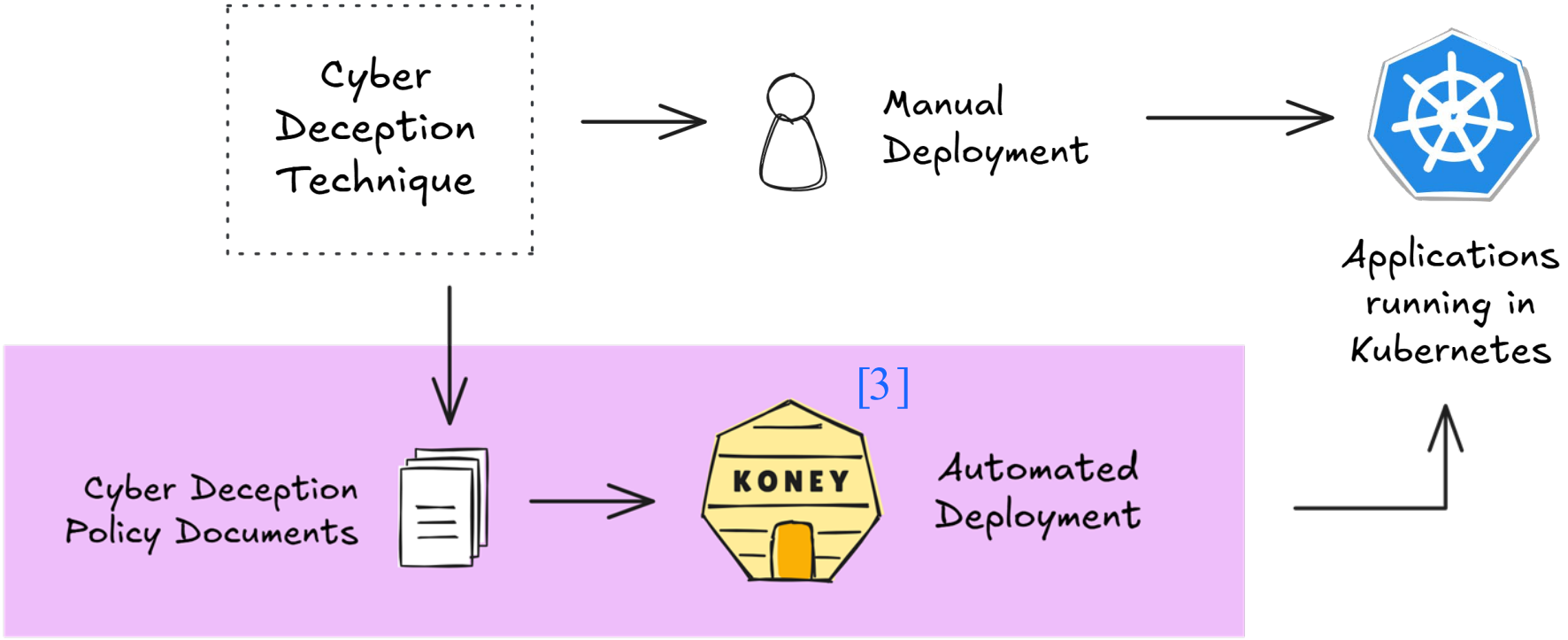


“Deception-As-Code” with Koney [3]

github.com/dynatrace-oss/koney

GitHub

…instead of manually deciphering techniques from academic papers.



[3] 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.

Cyber Deception Policy Documents

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 parameters

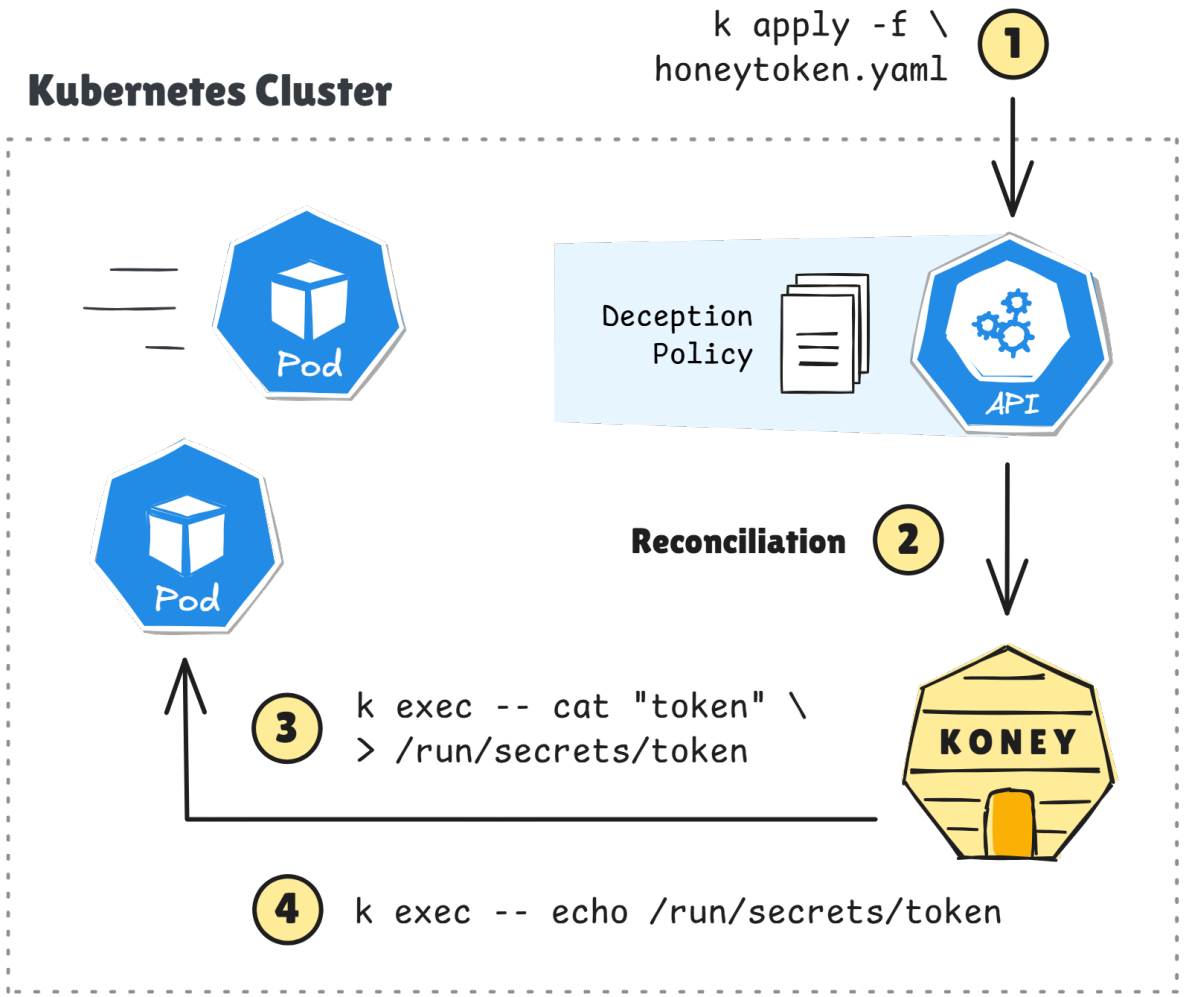
← Criteria for selecting the Kubernetes workloads (e.g., containers) in which to deploy the traps

← Decoy. [4] Strategy to deploy the trap itself

← Captor. [4] Strategy for monitoring the trap

[4] 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.

Placing Honeytokens by Executing Shell Commands

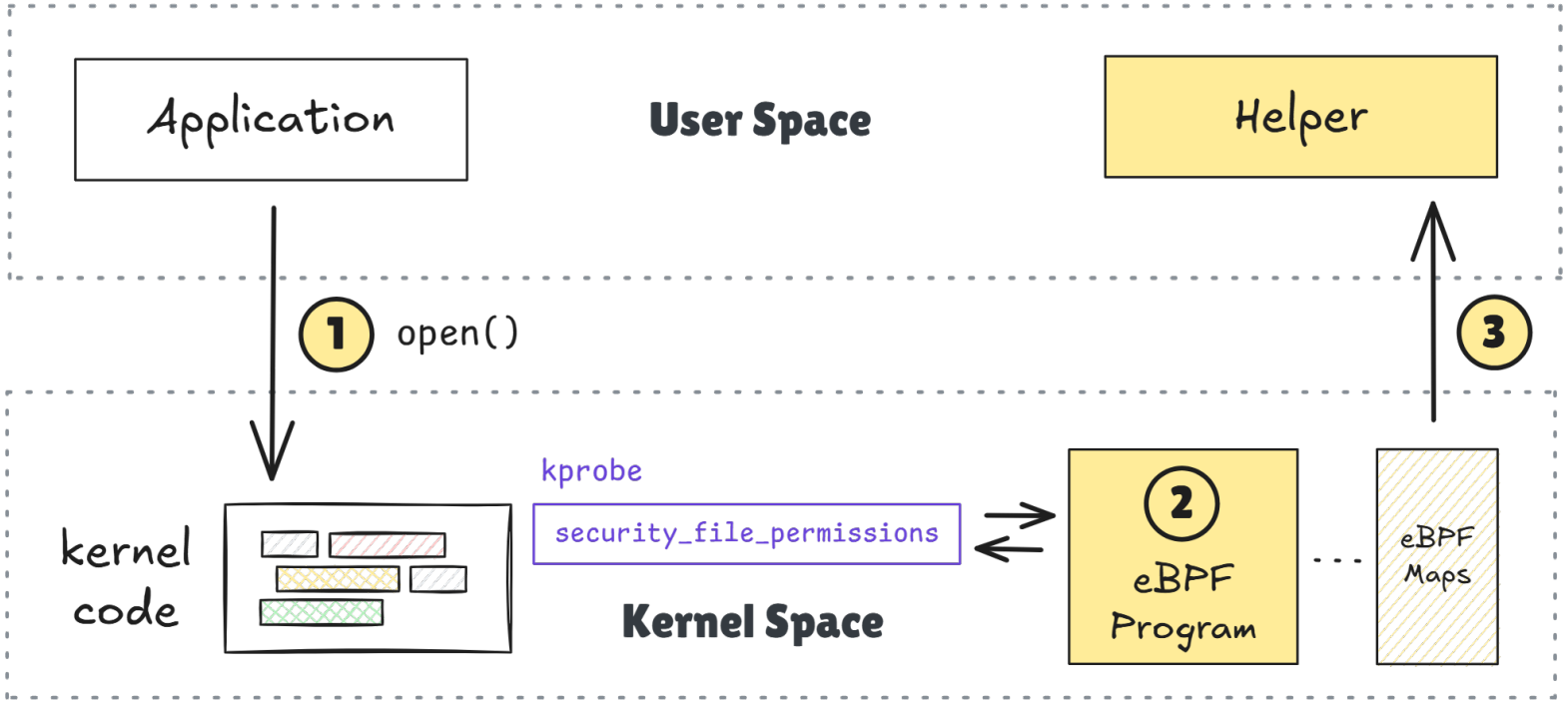


honeytoken.yaml — YAML

```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
```

File Access Monitoring with eBPF

"eBPF makes the kernel programmable."– We hook the kprobe security_file_permissions to detect honeytoken access attempts.



Koney uses Tetragon, which simplifies deploying eBPF programs in Kubernetes.

Koney Alert Example

alert.json JSON


```
{
  "timestamp": "2025-06-02T11:17:02Z",
  "deception_policy_name": "deceptionpolicy-servicetoken",
  "trap_type": "filesystem_honeytoken",
  "metadata": { "file_path": "/run/secrets/koney/service_token" },
  "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/koney/service_token"
  }
}
```

Beyond Honeypots

Application Layer Cyber Deception

Place honeytokens.
Add deceptive HTTP endpoints.
Modify HTTP headers and bodies.

Enticing Traps

“backup.tar.gz”, not “passwords.txt”.
Imitate true system components.

“Deception-As-Code”

Formalize traps with
policy documents.

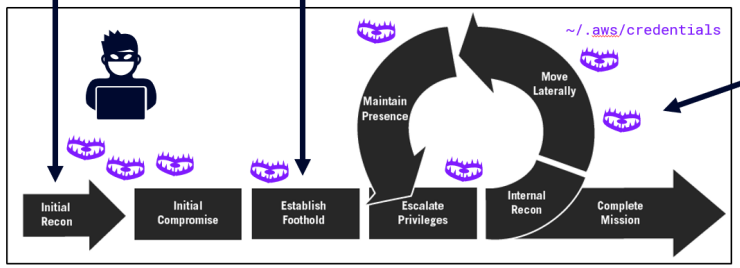
Deceive. Test. Repeat.

Deploy small traps.
Share insights.

What is Cyber Deception?

Attacks start with reconnaissance

Sophisticated adversaries will eventually find a way to infiltrate your network



[1]

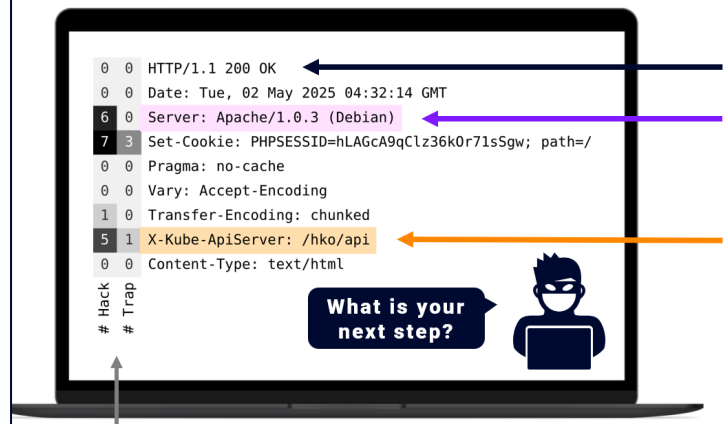
We can place honeytokens, install fake endpoints, and inject deceptive content to …

- 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].

The Honeyquest Study [2]

github.com/dynatrace-oss/honeyquest



Neutral Elements

Risky Element
e.g., a true indicator of vulnerability CVE-1999-0067

Deceptive Element
Traps injected by us

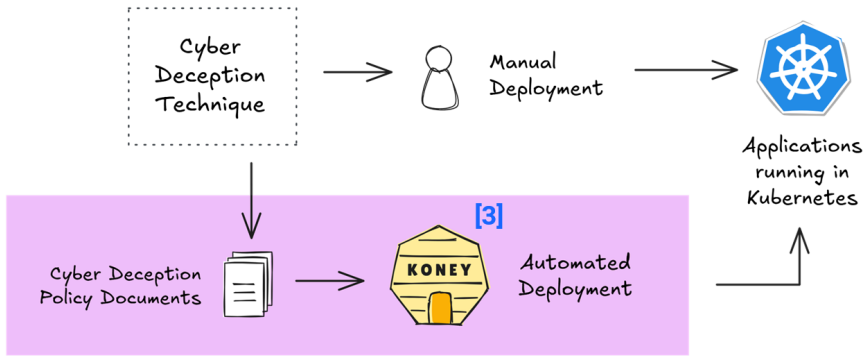
number of marks placed by participants (n=47)

[2] M. Kahlhofer, S. Achleitner, S. Rass, and R. Mayrhofer, "Honeyquest: Rapidly Measuring the Enticingness of Cyber Deception Techniques with Code-based Questionnaires," in RAID '24. Padua, Italy: ACM, 2024, pp. 317–336. doi: 10.1145/3678890.3678897.

"Deception-As-Code" with Koney [3]

github.com/dynatrace-oss/koney

… instead of manually deciphering techniques from academic papers.



[3] 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.

Beyond Honeypots

Application Layer Cyber Deception
Place honeytokens.
Add deceptive HTTP endpoints.
Modify HTTP headers and bodies.

Create Enticing Traps
"backup.tar.gz", not "passwords.txt".
Imitate true system components.

"Traps-As-Code"
Formalize traps with policy documents.

Deceive. Test. Repeat.
Deploy small traps.
Share insights.

CLOUD DONE RIGHT

APPENDIX

Adding Fake Routes to Web Applications

aws-credentials.yaml — YAML

```yaml
apiVersion: koney.io/v1alpha1
kind: DeceptionPolicy
spec:
  traps:
    - httpResponse:
        request:
          match: /.aws/credentials$  # ends with
          method: GET                # GET only
        response:
          status: 200
          headers:
            Content-Type: text/plain
          body: |
            aws_access_key_id = ASLPVFCMPNXCFEX
            aws_secret_access_key = h8aQcFM64
      match:
        any:
          - resources:
              ports: [80, 8080]
              selector:
                matchLabels:
                  op/aws-credentials: true
```

disallow-injection.yaml — YAML

```yaml
apiVersion: koney.io/v1alpha1
kind: DeceptionPolicy
spec:
  traps:
    - httpBodyMutation:
        request:
          match: ^/robots.txt$
          method: GET
        response:
          matchHeaders:
            Content-Type: text/html
          bodyMutations:
            - engine: regex
              match: (?s)(.*)
              replace: "$1\nDisallow: /.aws/credentials"
      match:
        any:
          - resources:
              selector:
                matchLabels:
                  op/disallow-injection: true
```

APPENDIX

File Access Monitoring with eBPF via Tetragon

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. [Online]. Available: https://isovalent.com/blog/post/file-monitoring-with-ebpf-and-tetragon-part-1/

APPENDIX

Related Projects

patrickpichler/**beesting**

1 Contributor · 0 Issues · 7 Stars · 0 Forks

San7o/**hive-operator**

Kubernetes operator for kernel tracing of inode accesses with eBPF programs

1 Contributor · 0 Issues · 5 Stars · 0 Forks

utkusen/**baitroute**

A web honeypot library to create vulnerable-looking endpoints to detect and mislead attackers

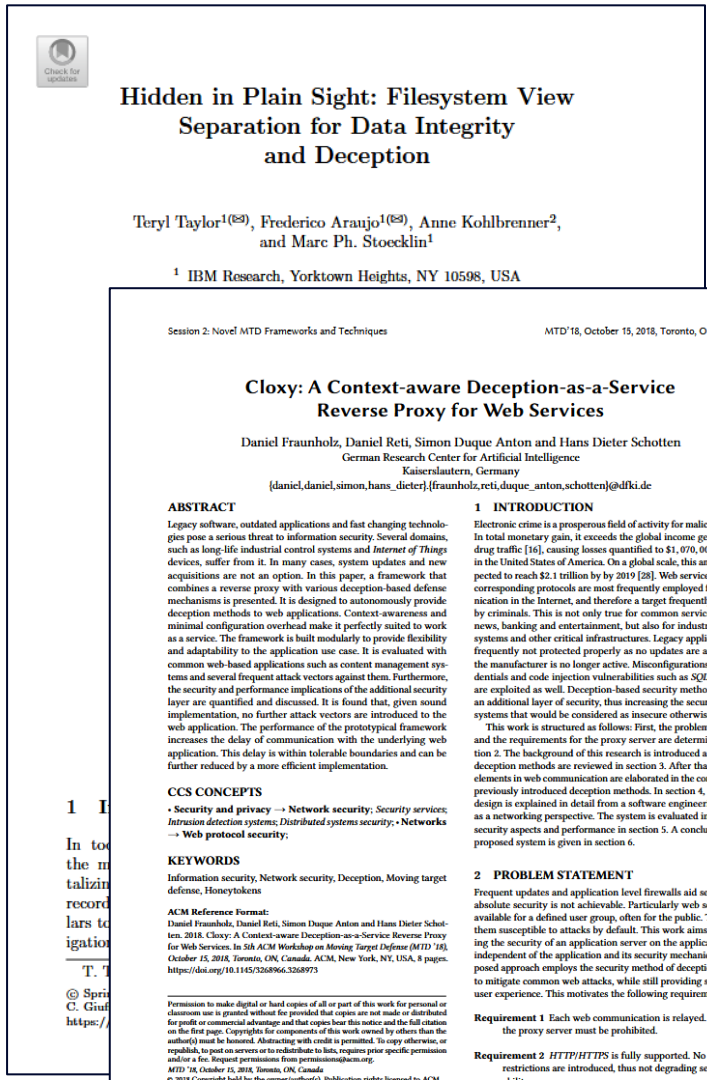
2 Contributors · 2 Issues · 404 Stars · 17 Forks

DataDog/**HASH**

HASH (HTTP Agnostic Software Honeypot)

6 Contributors · 2 Used by · 140 Stars · 8 Forks

[6] DcyFS



Hidden in Plain Sight: Filesystem View Separation for Data Integrity and Deception

Teryl Taylor[1(✉)], Frederico Araujo[1(✉)], Anne Kohlbrenner[2], and Marc Ph. Stoecklin[1]

[1] IBM Research, Yorktown Heights, NY 10598, USA

[7] Cloxy

Session 2: Novel MTD Frameworks and Techniques — MTD'18, October 15, 2018, Toronto, ON, Canada

Cloxy: A Context-aware Deception-as-a-Service Reverse Proxy for Web Services

Daniel Fraunholz, Daniel Reti, Simon Duque Anton and Hans Dieter Schotten
German Research Center for Artificial Intelligence
Kaiserslautern, Germany
{daniel,daniel,simon,hans_dieter}.{fraunholz,reti,duque_anton,schotten}@dfki.de

ABSTRACT

Legacy software, outdated applications and fast changing technologies pose a serious threat to information security. Several domains, such as long-life industrial control systems and *Internet of Things* devices, suffer from it. In many cases, system updates and new acquisitions are not an option. In this paper, a framework that combines a reverse proxy with various deception-based defense mechanisms is presented. It is designed to autonomously provide deception methods to web applications. Context-awareness and minimal configuration overhead make it perfectly suited to work as a service. The framework is built modularly to provide flexibility and adaptability to the application use case. It is evaluated with common web-based applications such as content management systems and several frequent attack vectors against them. Furthermore, the security and performance implications of the additional security layer are quantified and discussed. It is found that, given sound implementation, no further attack vectors are introduced to the web application. The performance of the prototypical framework increases the delay of communication with the underlying web application. This delay is within tolerable boundaries and can be further reduced by a more efficient implementation.

CCS CONCEPTS

• **Security and privacy** → **Network security**; *Security services*; *Intrusion detection systems*; *Distributed systems security*; • **Networks** → **Web protocol security**;

KEYWORDS

Information security, Network security, Deception, Moving target defense, Honeytokens

ACM Reference Format:

1 INTRODUCTION

Electronic crime is a prosperous field of activity for malicious actors. In total monetary gain, it exceeds the global income generated by drug traffic [16], causing losses quantified to $1,070,000,000 [17] in the United States of America. On a global scale, this amount is expected to reach $2.1 trillion by by 2019 [28]. Web services and their corresponding protocols are most frequently employed for communication in the Internet, and therefore a target frequently exploited by criminals. This is not only true for common services, such as news, banking and entertainment, but also for industrial control systems and other critical infrastructures. Legacy applications are frequently not protected properly as no updates are available or the manufacturer is no longer active. Misconfigurations, weak credentials and code injection vulnerabilities such as *SQL* injections are exploited as well. Deception-based security methods can add an additional layer of security, thus increasing the security level of systems that would be considered as insecure otherwise.

This work is structured as follows: First, the problem is defined and the requirements for the proxy server are determined in section 2. The background of this research is introduced and defense deception methods are reviewed in section 3. After that, common elements in web communication are elaborated in the context of the previously introduced deception methods. In section 4, the system design is explained in detail from a software engineering as well as a networking perspective. The system is evaluated in respect to security aspects and performance in section 5. A conclusion of the proposed system is given in section 6.

2 PROBLEM STATEMENT

Frequent updates and application level firewalls aid security, but absolute security is not achievable. Particularly web services are available for a defined user group, often for the public. This makes them susceptible to attacks by default. This work aims at increasing the security of an application server on the application layer, independent of the application and its security mechanics. The proposed approach employs the security method of deception in order to mitigate common web attacks, while still providing satisfactory user experience. This motivates the following requirements:

Requirement 1 Each web communication is relayed. Bypassing the proxy server must be prohibited.

Requirement 2 *HTTP/HTTPS* is fully supported. No functional restrictions are introduced, thus not degrading service availability.

[6] T. Taylor, F. Araujo, A. Kohlbrenner, and M. Ph. Stoecklin, "Hidden in Plain Sight: Filesystem View Separation for Data Integrity and Deception," in Detection of Intrusions and Malware, and Vulnerability Assessment, C. Giuffrida, S. Bardin, and G. Blanc, Eds., in DIMVA '18. Saclay, France: Springer International Publishing, Jun. 2018, pp. 256–278. doi: 10.1007/978-3-319-93411-2_12.

[7] D. Fraunholz, D. Reti, S. Duque Anton, and H. D. Schotten, "Cloxy: A Context-aware Deception-as-a-Service Reverse Proxy for Web Services," in Proceedings of the 5th ACM Workshop on Moving Target Defense, in MTD '18. Toronto, Canada: Association for Computing Machinery, Jan. 2018, pp. 40–47. doi: 10.1145/3268966.3268973.