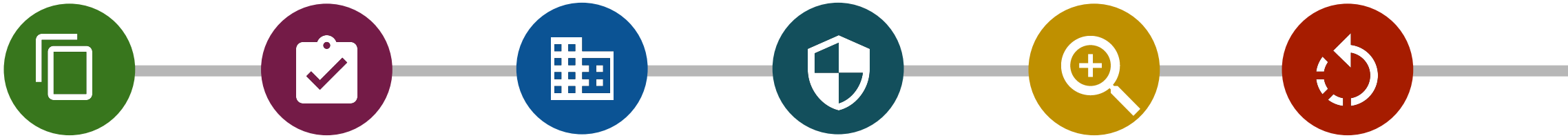
Shopify's $25K Bug Report

and the cluster takeover that didn't happen

Greg Castle
GKE Security Tech Lead
Twitter: @mrgcastle
Github: @destijl
Google

Shane Lawrence
Security Infrastructure Engineer
Twitter: @shaneplawrence
Github: @shane-lawrence
Shopify

A production security story



Introduction Bug report Exchange Attack & defense Detection Takeaways

Introduction

Shopify’s background

Who is shopify.com?

2017 data

600K+
merchants

26B+
$processed

80K+
requests per second
peak traffic

shopify

Shopify cloud platform

- Scalable
- Application developers don’t need to learn k8
- Self-serve with guardrails & paved roads
- Security*** by default

shopify

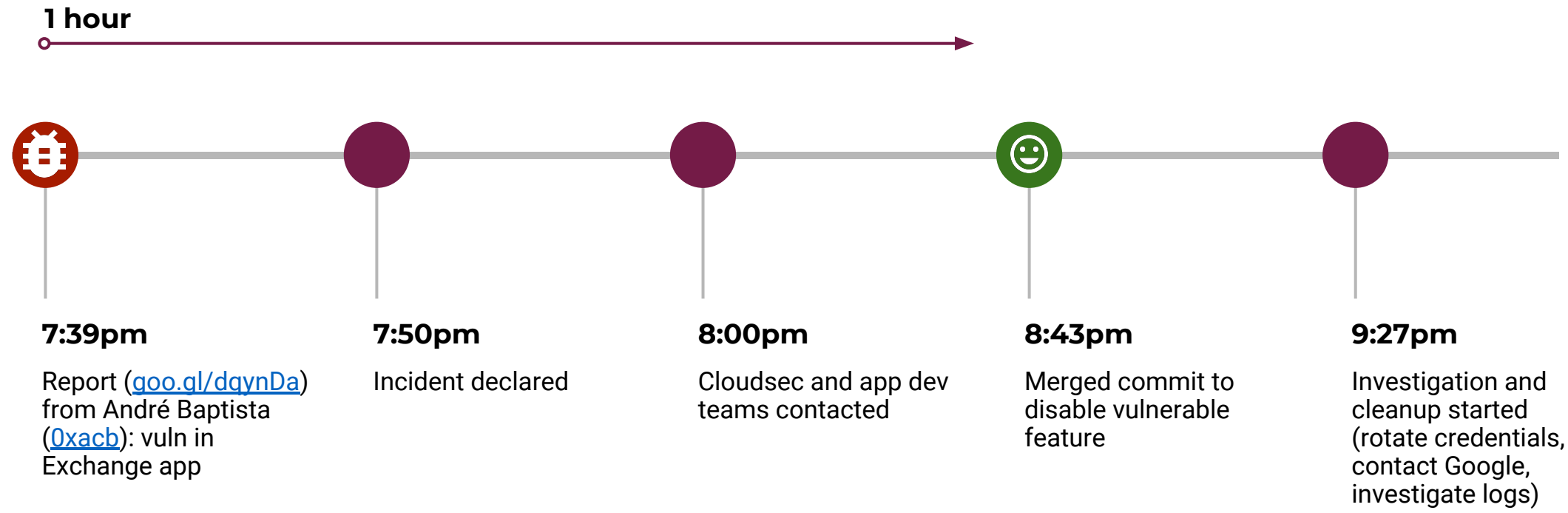
Shopify’s bug bounty programs

- 330+ hackers over 3+ years
- Merchants and buyers protected
- **$1,000,000+ paid**
- hackerone.com/shopify

shopify

Bug report

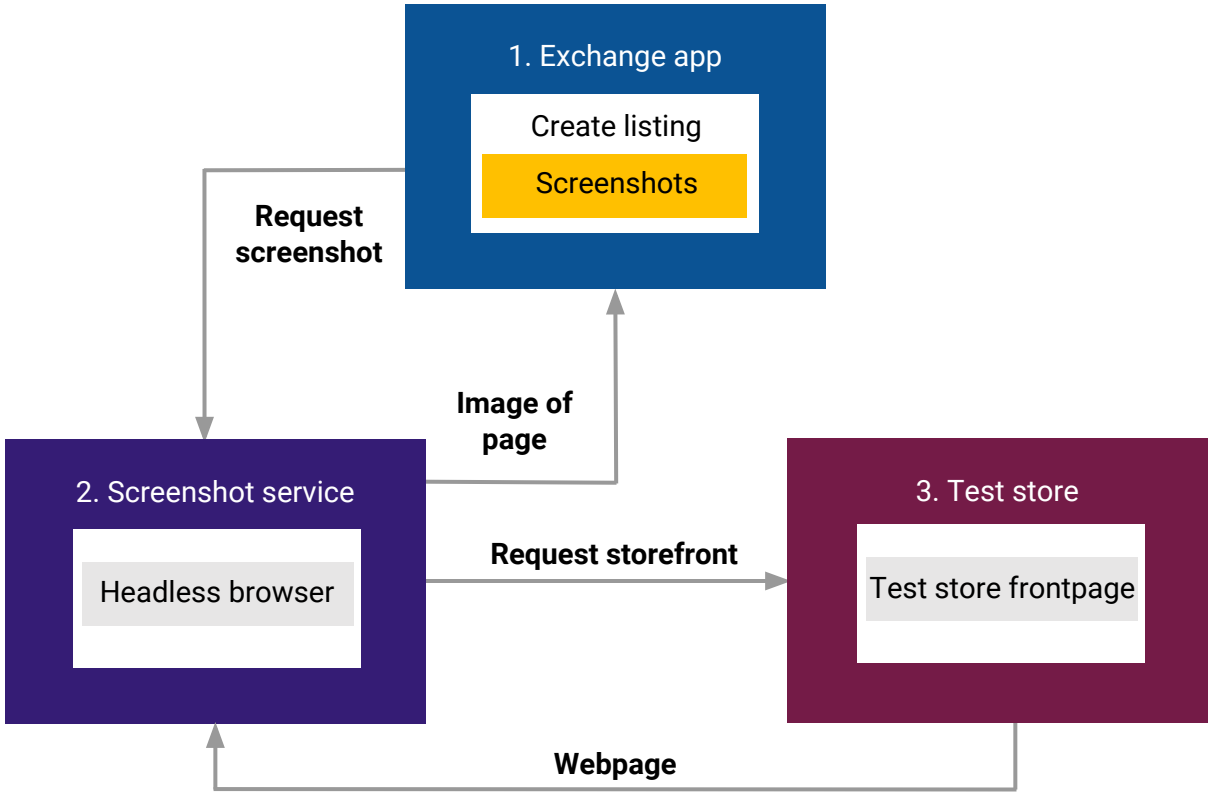
Security report and responses



Exchange

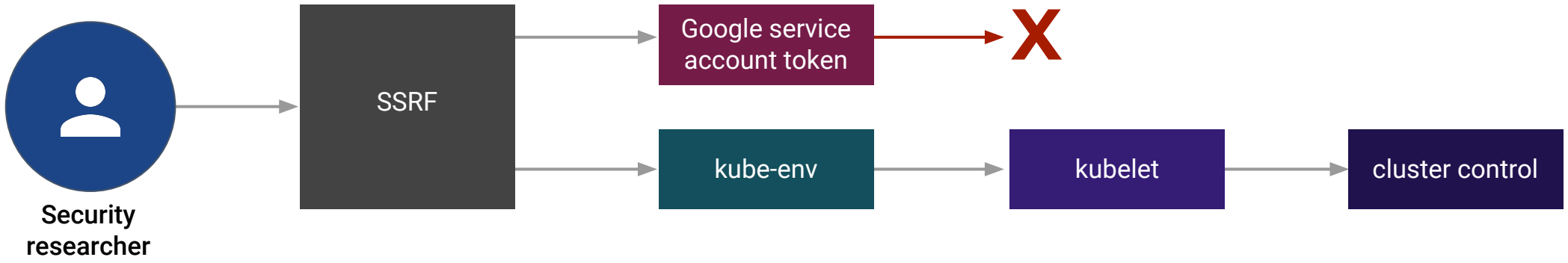
What is Exchange?

Marketplace for buying & selling stores

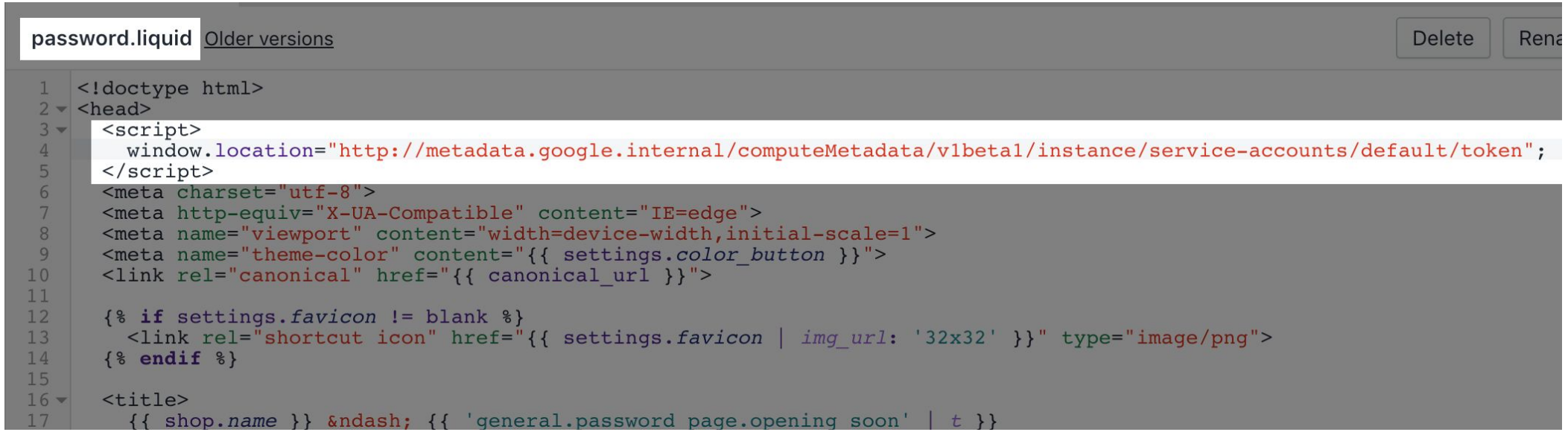


Attack & defense

The attack

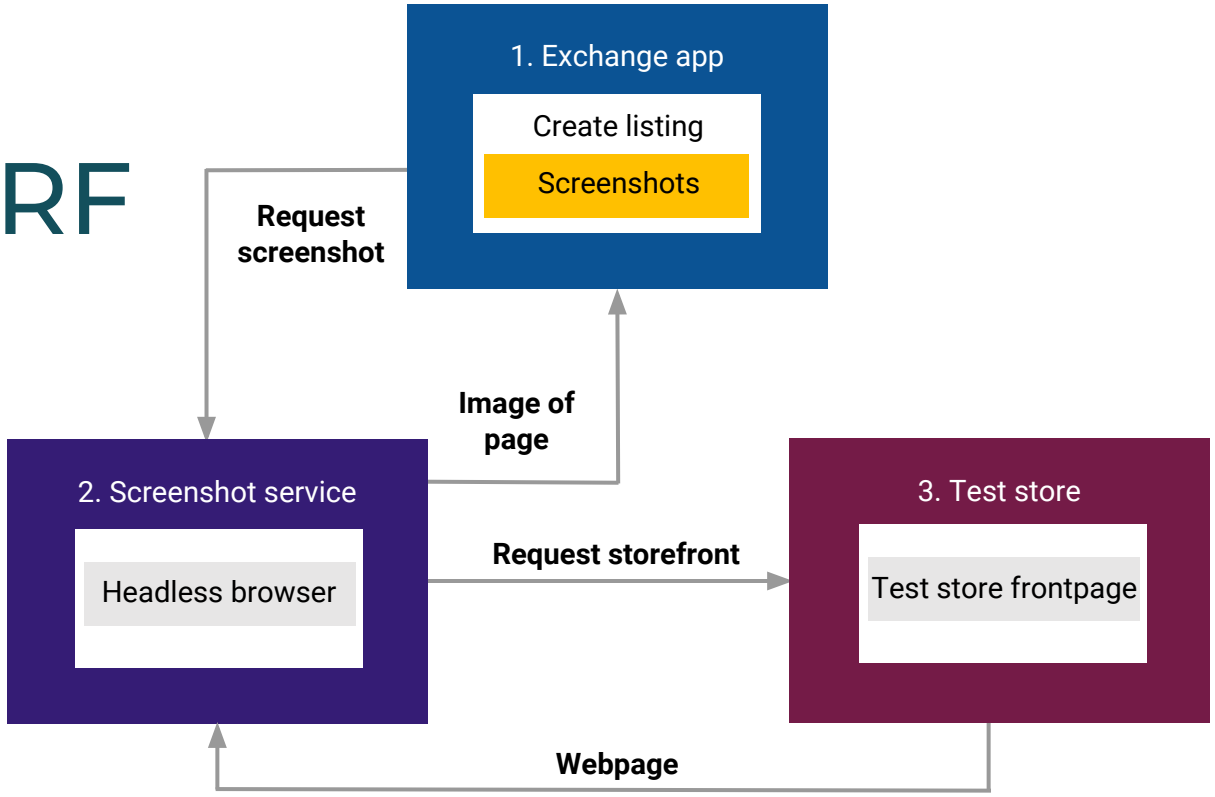

Security researcher
SSRF
Google service account token
X
kube-env
kubelet
cluster control

Server Side Request Forgery (SSRF)



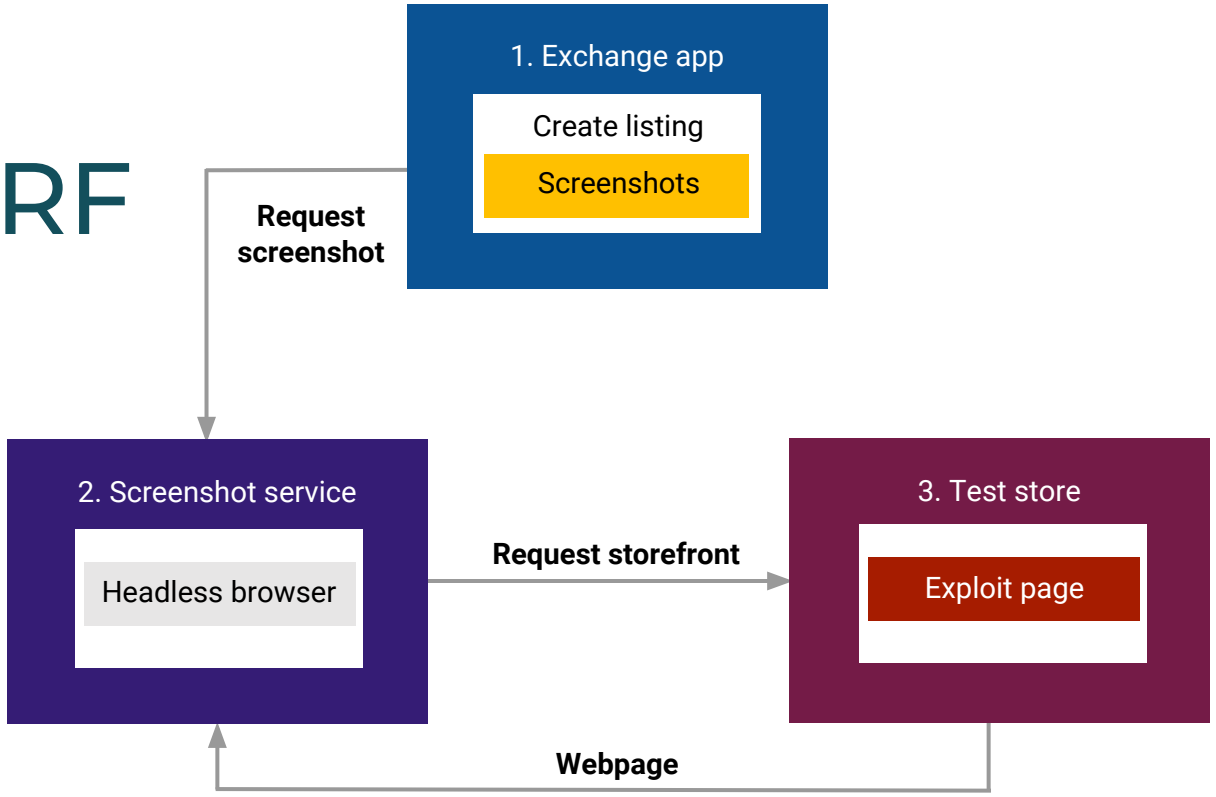
Attack: Weaponize SSRF

Existing workflow



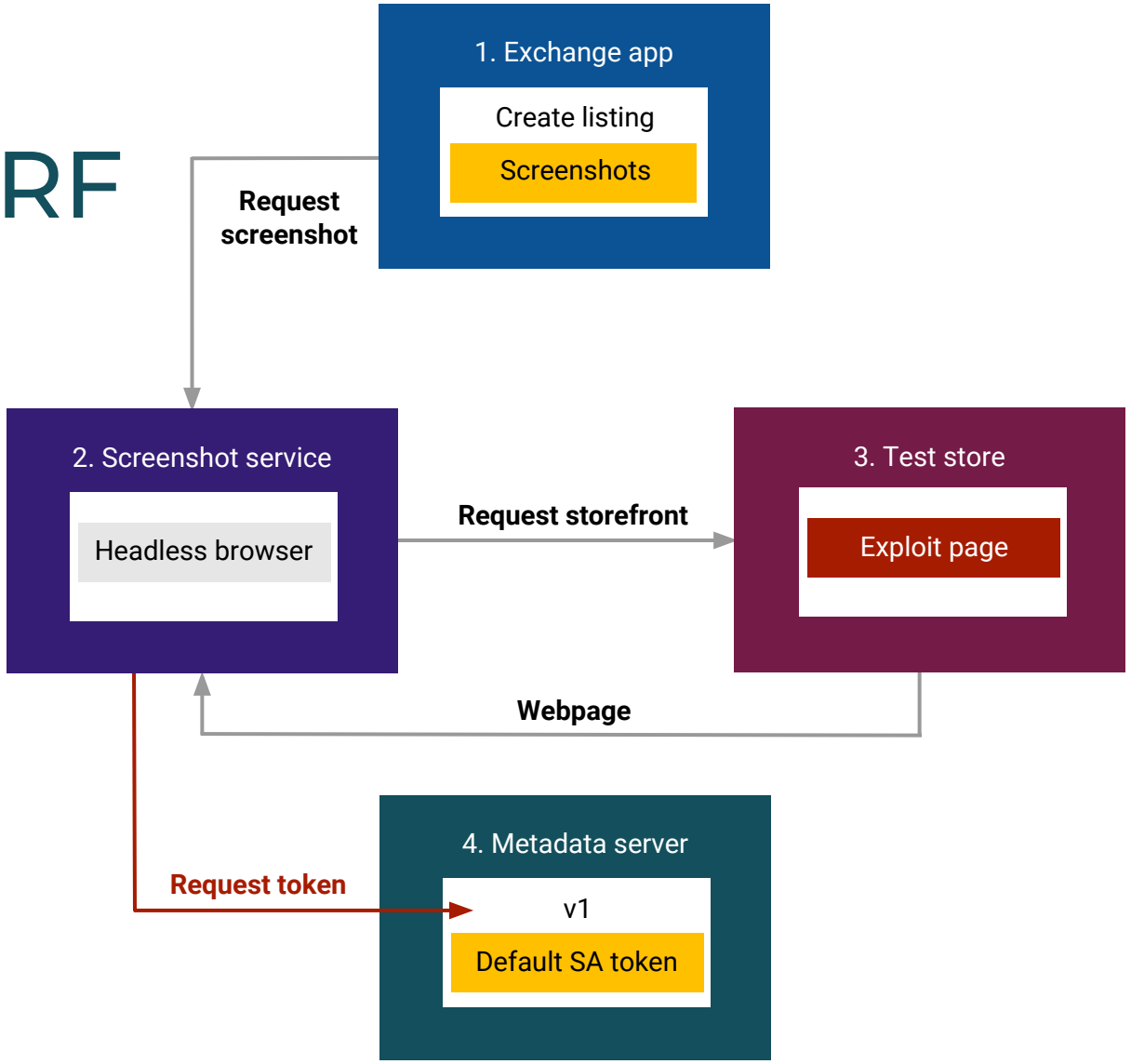
Attack: Weaponize SSRF

Got token for the VM's Google service account

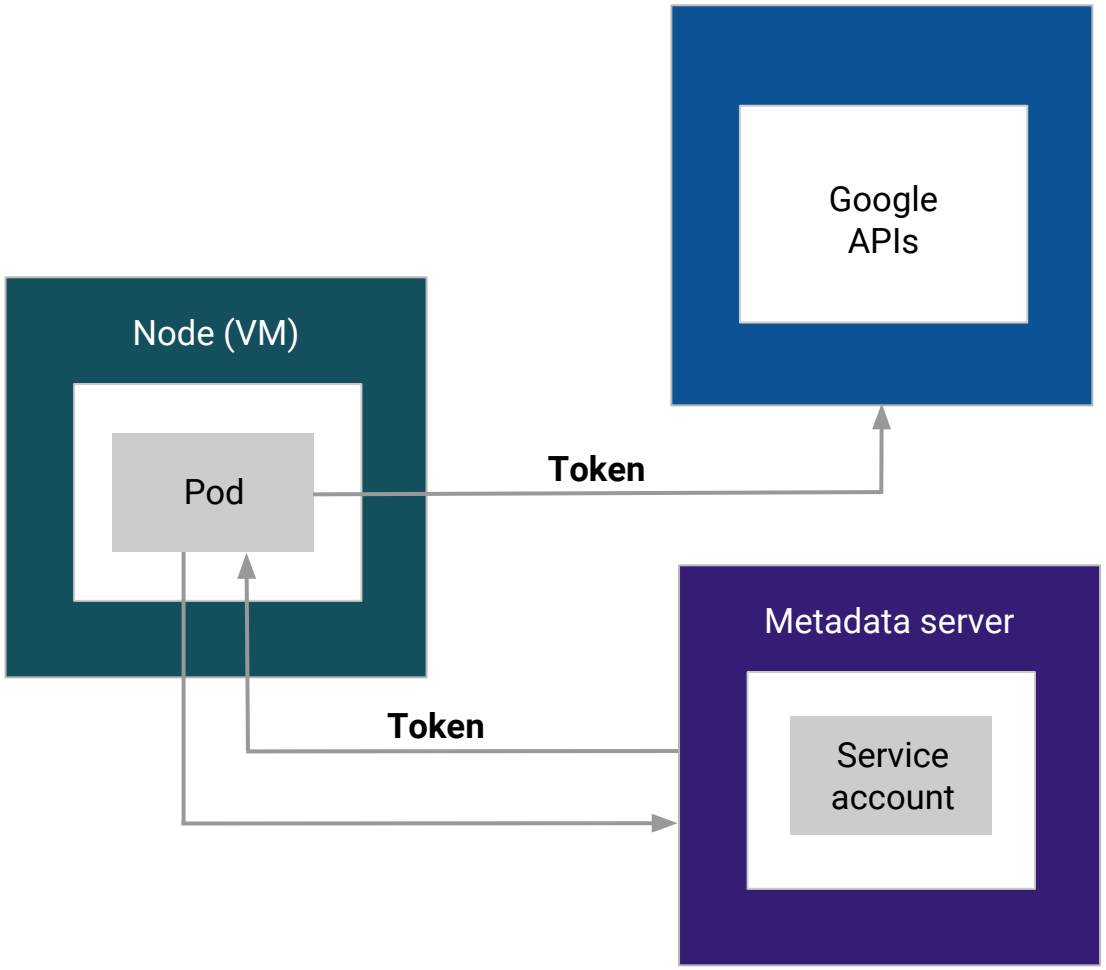


Attack: Weaponize SSRF

Got token for the VM's Google service account



Sidebar: What is this Google SA?

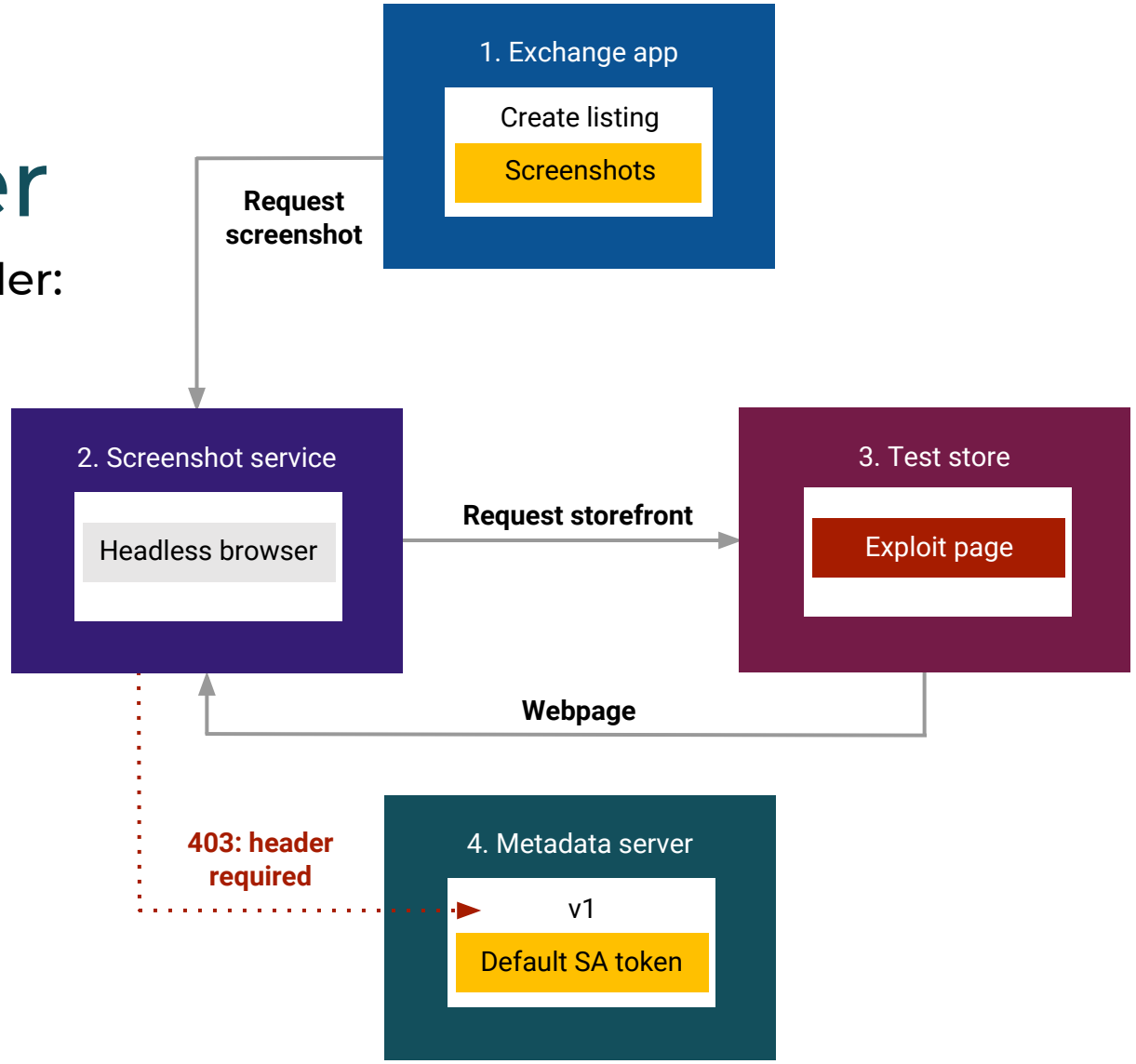


Demo

Token attack

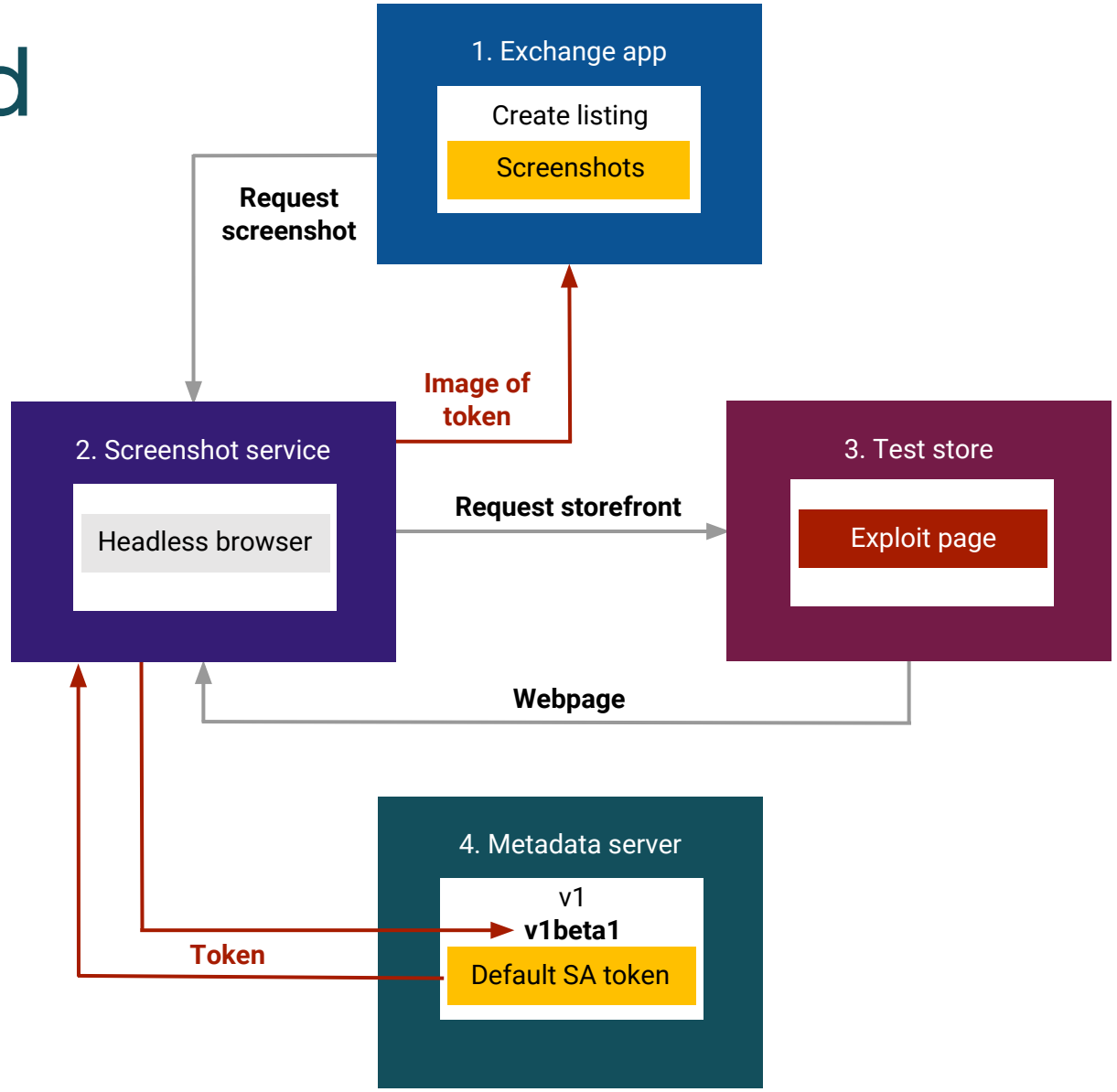
Defense: Require header

Metadata server requires header:
Metadata-Flavor: Google



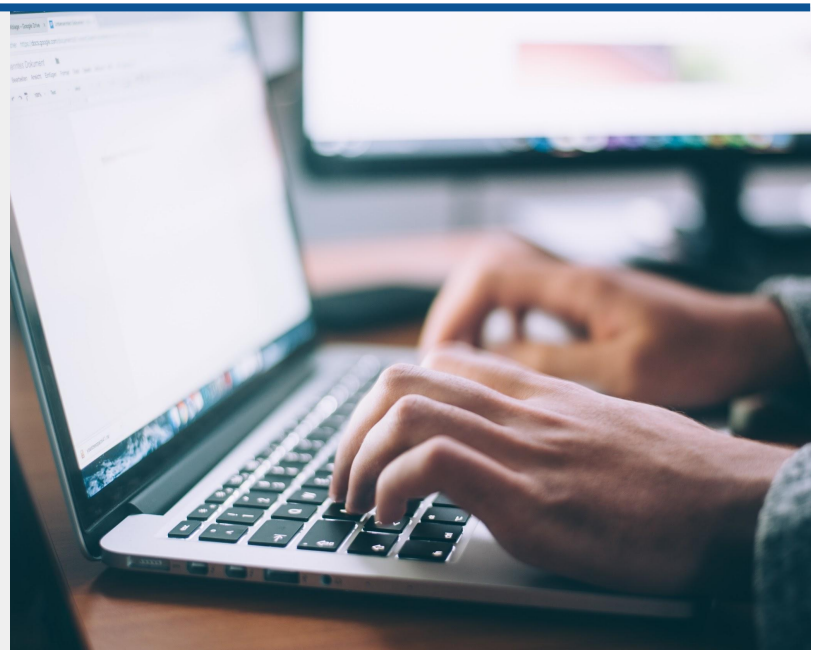
Attack: Use old API version

Beta API: no request header required ☹



Defense: Disable old API versions

- Beta API known issue: APIs still in use
- Disabled by default in new 1.12+ clusters
- Opt-in now: “disable-legacy-endpoints=true”
- goo.gl/JsdJbL for details



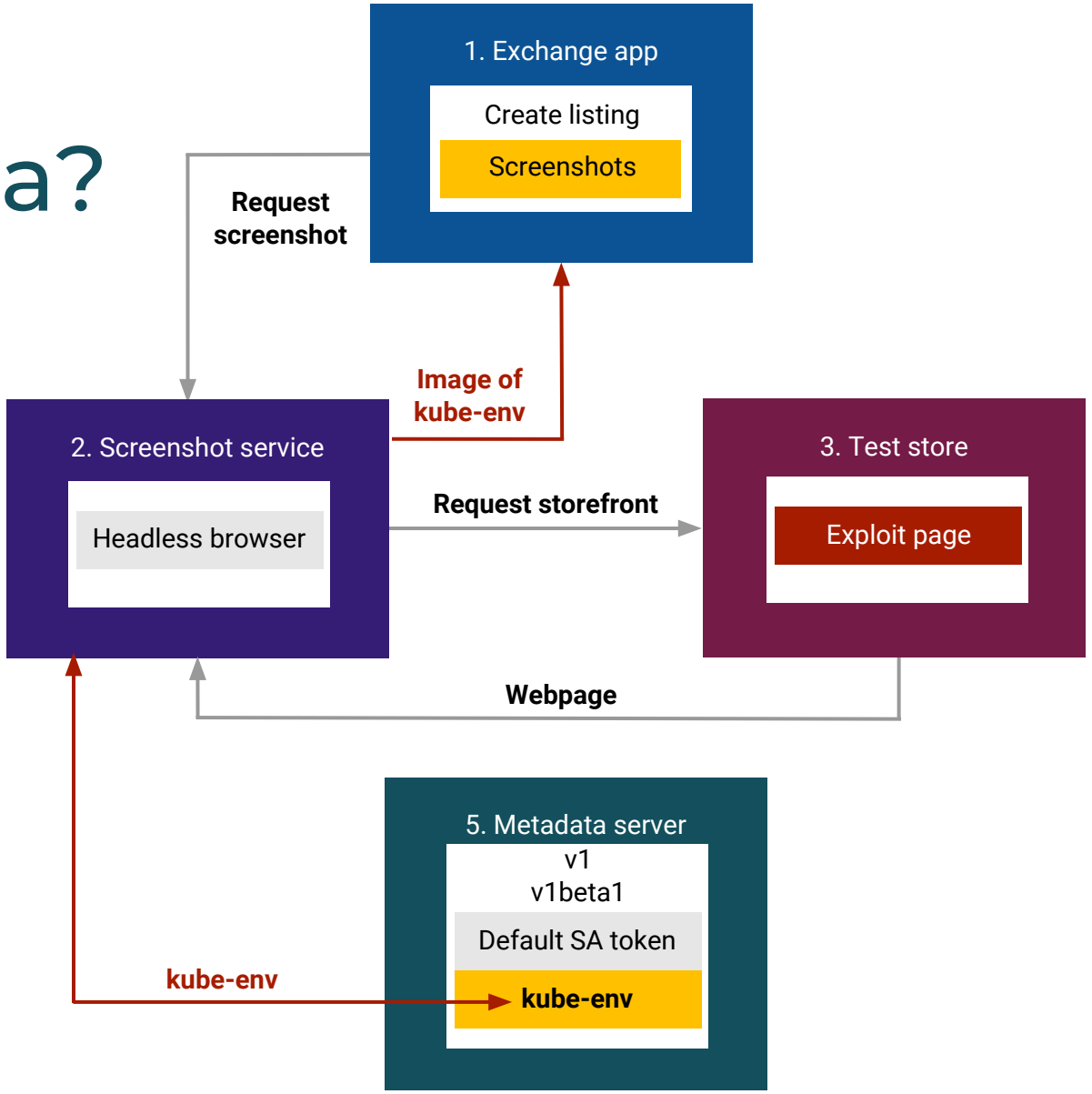
Defense: Least priv on token

- Default SA least privilege from 1.10+
- May vary if users have granted extra privs
- Shopify had minimal privs for log/mon/debug
- Token not useful to researcher

Attack: What other metadata?

Metadata server = trust bootstrap for nodes

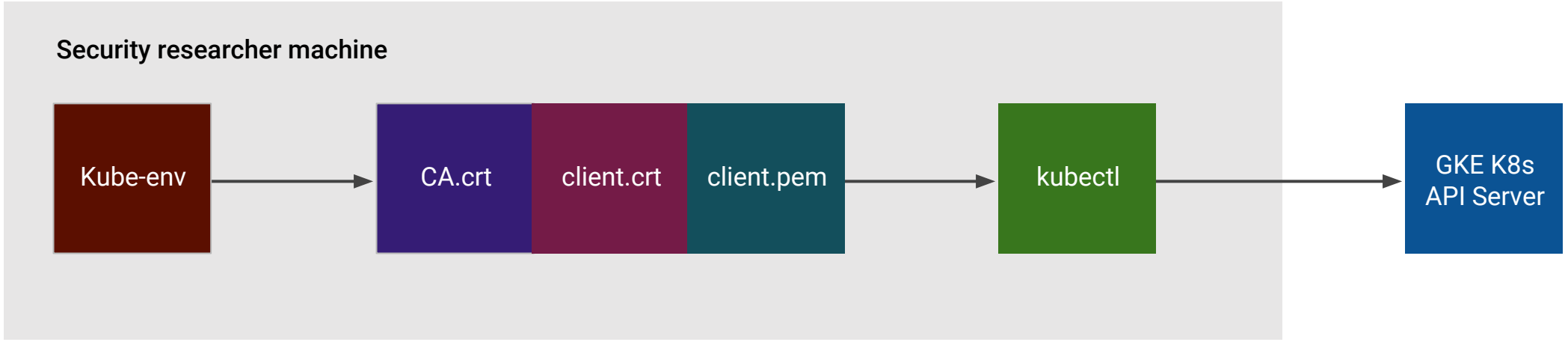
Export static key from “kube-env”



Demo

kube-env attack

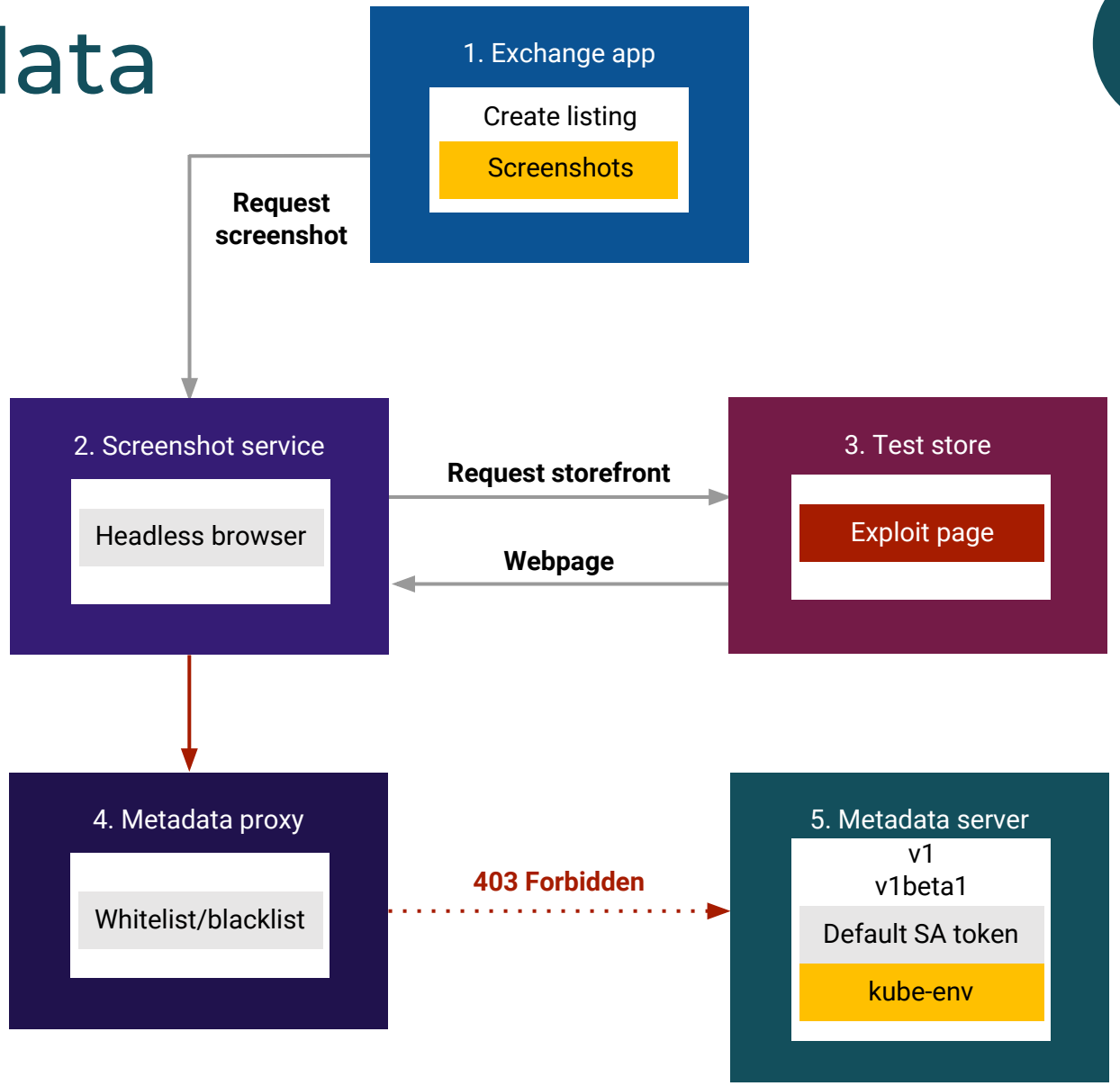
Attack: Kubelet bootstrap key



Defense: Metadata concealment

Now: metadata concealment (Beta) goo.gl/u6rrMT

Future: K8s TPM trust bootstrap



Defense: Minimize kubelet privs

- RBAC on (ABAC off): GKE default
- Node Authorization on: GKE default
- Audit role bindings:
  GKE “kubelet-cluster-admin” (not actually cluster admin) binding if upgraded cluster

https://kubernetes.io/docs/tasks/administer-cluster/securing-a-cluster/

Demo

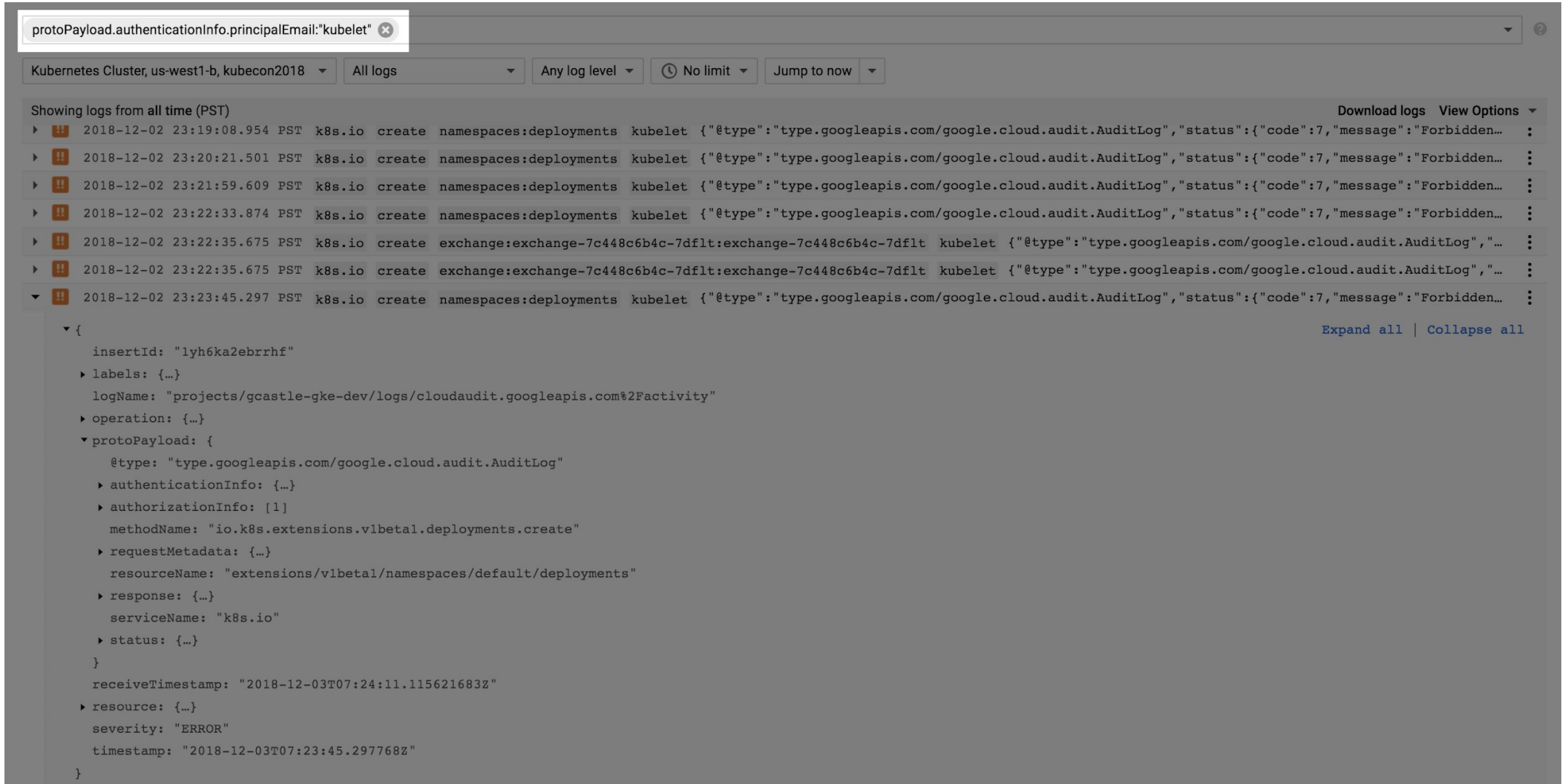
Defenses

Detection

What's in the logs?

- K8s API audit logs: goo.gl/d8YebH
- Content depends on audit policy
- GKE: g.co/gke/auditlogging

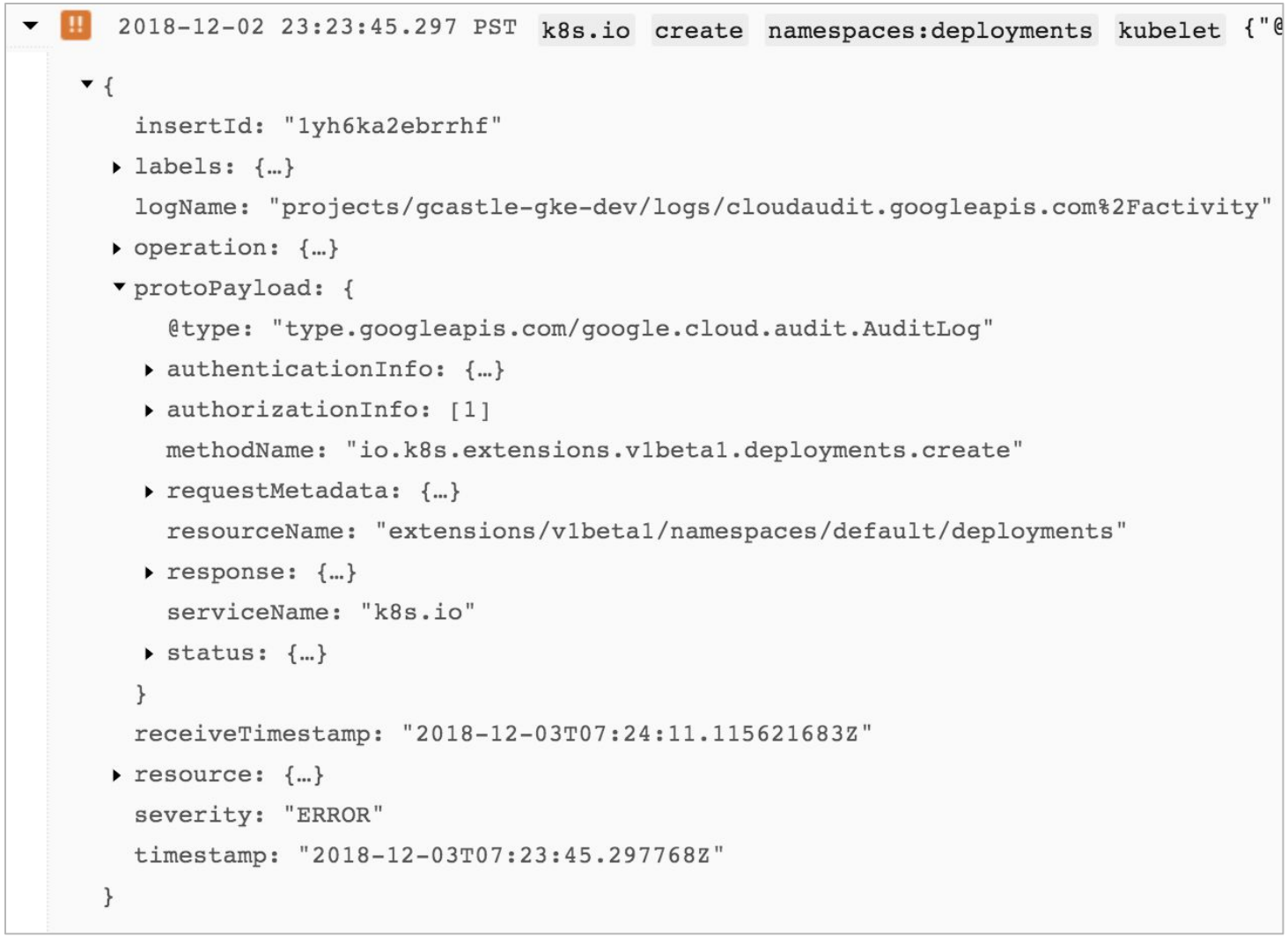
Filter logs for kubelet user



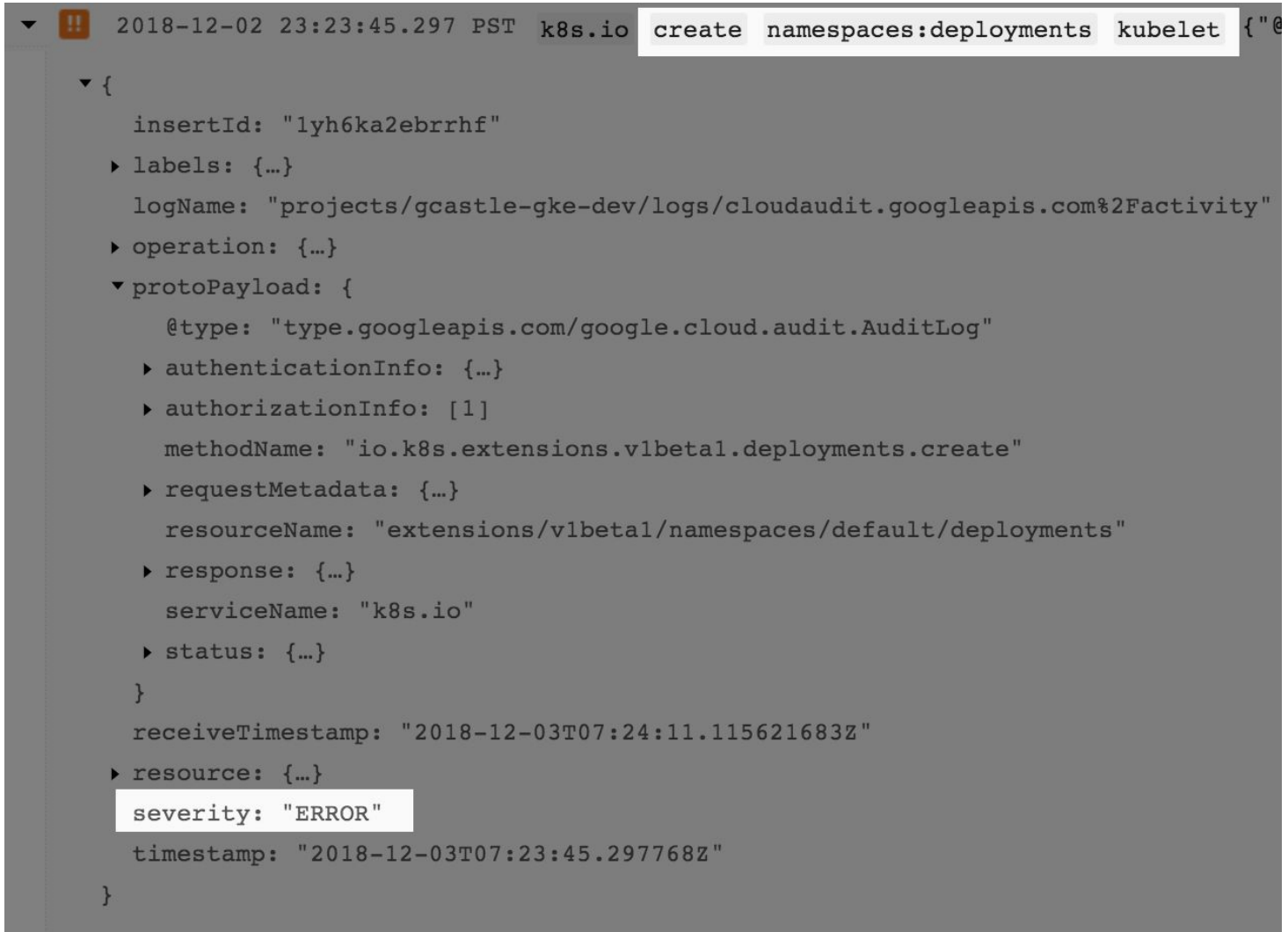
Filter logs for kubelet user

```
protoPayload.authenticationInfo.principalEmail:"kubelet"
```

Create deployment failed



Create deployment



Exec into exchange pod

2018-12-02 23:23:48.027 PST `k8s.io` `create` `exchange:exchange-7c448c6b4c-7dflt:exchange-7c448c6b4c-7dflt` `kubelet`

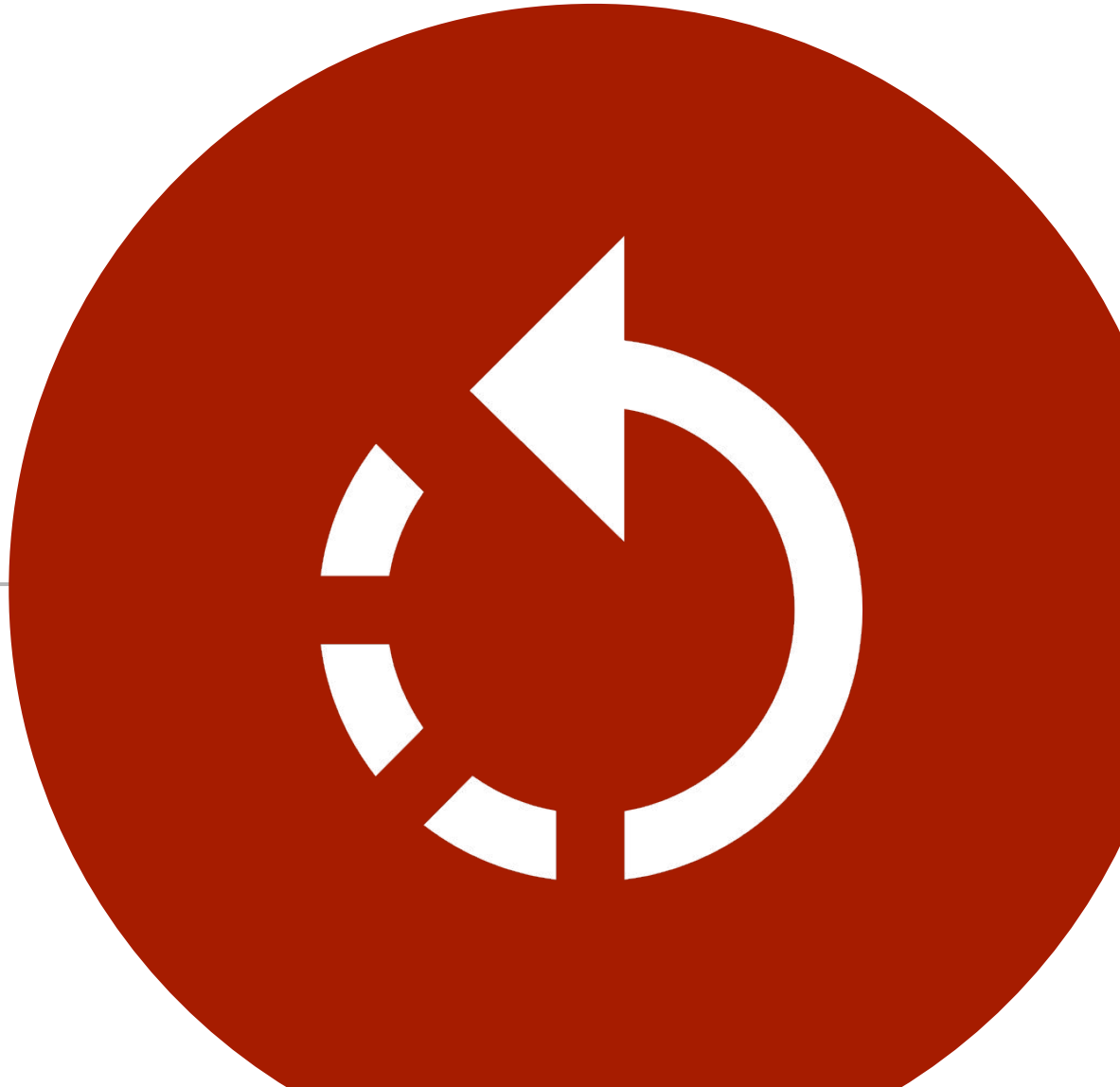

```
{
  insertId: "1yh6ka2ebrrhk"
  labels: {…}
  logName: "projects/gcastle-gke-dev/logs/cloudaudit.googleapis.com%2Factivity"
  operation: {…}
  protoPayload: {
    @type: "type.googleapis.com/google.cloud.audit.AuditLog"
    authenticationInfo: {…}
    authorizationInfo: [1]
    methodName: "io.k8s.core.v1.pods.exec.create"
    requestMetadata: {…}
    resourceName: "core/v1/namespaces/exchange/pods/exchange-7c448c6b4c-7dflt/exec/exchange-7c448c6b4c-7dflt"
    response: {…}
    serviceName: "k8s.io"
    status: {…}
  }
  receiveTimestamp: "2018-12-03T07:24:11.115621683Z"
  resource: {…}
  severity: "ERROR"
  timestamp: "2018-12-03T07:23:48.027080Z"
}
```

Exec into exchange pod

Node CSR creation

▸ i	2018-11-27 17:56:55.277 PST	k8s.io	create	node-csr-MsMEIvsQJpR6LrdDdMBNaKSS_oBqWQPBI-gIqgsM2P0	kubelet
▸ i	2018-11-27 17:56:55.368 PST	k8s.io	create	node-csr-LWM2CQ8kwDJ0vU1cRN1mF-NbsDJiMrqWtha8FpT68l4	kubelet
▸ i	2018-11-27 17:56:57.162 PST	k8s.io	create	node-csr-BDsWtKIBEX0mPw8cy6oE-2AdPCOD4MzQk-umBKukKJM	kubelet

Takeaways



Shopify’s response

1 day

- Disable vulnerable service.
- Start rotating credentials.
- Pay $ to researcher.

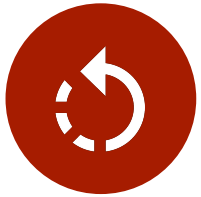
1 week

- Analyze audit logs.
- Clean up RBAC.

1 month

- Prevent unwanted redirects.
- Re-enable screenshot service.
- Deploy metadata proxy.
- Pay $$$ to researcher.
- Disclose vulnerability.

Lessons learned: K8s advice



- Follow cloud provider hardening advice (GKE: g.co/gke/hardening)
- Block off/filter access to any privileged network endpoints
- Run RBAC and Node Authorization (GKE default)
- Apply least privilege for K8s service accounts
- Audit role bindings, especially upgraded clusters
- Collect API logs and have them available to query (GKE default)

Links and references

Shopify bug bounty: hackerone.com/shopify

Bug report details: goo.gl/dqynDa

GKE disable old APIs: goo.gl/JsdJbL

GKE metadata conceal: goo.gl/u6rrMT

K8s API audit logs: goo.gl/d8YebH

GKE logging: g.co/gke/auditlogging

Greg Castle
GKE Security Tech Lead
Twitter: @mrgcastle
Github: @destijl
Google

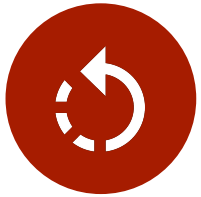
Shane Lawrence
Security Infrastructure Engineer
Twitter: @shaneplawrence
Github: @shane-lawrence
Shopify

Thank you

Reference

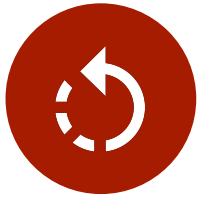
Log queries

Example log queries



- Broad strokes to get you started
- No standard language for queries like this
- SQL seems most standard
- But includes some BigQuery-isms for unpacking repeated fields
- Validation/tweaking on production clusters needed
- Mostly intended to point out interesting values and fields

RBAC Changes (excl system)

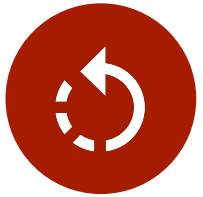


```
SELECT
  timestamp,
  protopayload_auditlog.methodName AS method,
  protopayload_auditlog.resourceName AS resource,
  protopayload_auditlog.authenticationInfo.principalEmail AS suid,
  authzinfo.granted AS granted,
  protopayload_auditlog.requestMetadata.callerIp AS saddr
FROM
`gcastle-gke-dev.kubecon2018.cloudaudit_googleapis_com_activity_*`,
UNNEST(protopayload_auditlog.authorizationInfo) AS authzinfo
WHERE
  protopayload_auditlog.methodName LIKE " io.k8s.authorization.rbac.v1%"
  AND NOT protopayload_auditlog.authenticationInfo.principalEmail LIKE "  system:%"
LIMIT 100
```

Similarly, use these methodName strings for specific changes to roles or bindings:

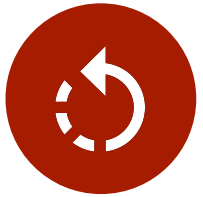
```
“io.k8s.authorization.rbac.v1.roles.%”
“io.k8s.authorization.rbac.v1.rolebindings.%”
“io.k8s.authorization.rbac.v1.clusterroles.%”
“io.k8s.authorization.rbac.v1.clusterrolebindings.%”
```

Creating CSRs via K8s API



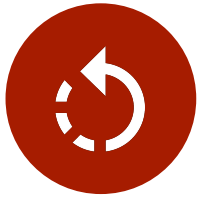
```
SELECT
  timestamp,
  protopayload_auditlog.methodName AS method,
  protopayload_auditlog.resourceName AS resource,
  protopayload_auditlog.authenticationInfo.principalEmail AS suid,
  authzinfo.granted AS granted,
  protopayload_auditlog.requestMetadata.callerIp AS saddr
FROM
`gcastle-gke-dev.kubecon2018.cloudaudit_googleapis_com_activity_*`,
UNNEST(protopayload_auditlog.authorizationInfo) AS authzinfo
WHERE
  protoPayload_auditlog.resourceName LIKE
"certificates.k8s.io/v1beta1/certificatesigningrequests%"
LIMIT 100
```

Unauth’d web requests



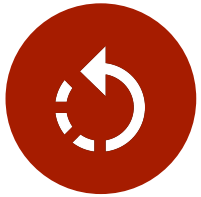
```
SELECT
  timestamp,
  protopayload_auditlog.methodName AS method,
  protopayload_auditlog.resourceName AS resource,
  protopayload_auditlog.authenticationInfo.principalEmail AS suid,
  authzinfo.granted AS granted,
  protopayload_auditlog.requestMetadata.callerIp AS saddr
FROM
`gcastle-gke-dev.kubecon2018.cloudaudit_googleapis_com_activity_*`,
UNNEST(protopayload_auditlog.authorizationInfo) AS authzinfo
WHERE
  protopayload_auditlog.authenticationInfo.principalEmail = " system:anonymous"
LIMIT 100
```

Kubelet bootstrap identity calls (GKE specific)



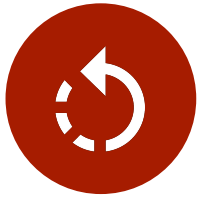
```
SELECT
  timestamp,
  protopayload_auditlog.methodName AS method,
  protopayload_auditlog.resourceName AS resource,
  protopayload_auditlog.authenticationInfo.principalEmail AS suid,
  authzinfo.granted AS granted,
  protopayload_auditlog.requestMetadata.callerIp AS saddr
FROM
`gcastle-gke-dev.kubecon2018.cloudaudit_googleapis_com_activity_*`,
UNNEST(protopayload_auditlog.authorizationInfo) AS authzinfo
WHERE protopayload_auditlog.authenticationInfo.principalEmail LIKE " kubelet"
LIMIT 100
```

Node authenticated requests



```
SELECT
  timestamp,
  protopayload_auditlog.methodName AS method,
  protopayload_auditlog.resourceName AS resource,
  protopayload_auditlog.authenticationInfo.principalEmail AS suid,
  authzinfo.granted AS granted,
  protopayload_auditlog.requestMetadata.callerIp AS saddr
FROM
`gcastle-gke-dev.kubecon2018.cloudaudit_googleapis_com_activity_*`,
UNNEST(protopayload_auditlog.authorizationInfo) AS authzinfo
WHERE protopayload_auditlog.authenticationInfo.principalEmail LIKE " system:node%"
LIMIT 100
```

Calls outside IP range



```
SELECT
  timestamp,
  protopayload_auditlog.methodName AS method,
  protopayload_auditlog.resourceName AS resource,
  protopayload_auditlog.authenticationInfo.principalEmail AS suid,
  authzinfo.granted AS granted,
  protopayload_auditlog.requestMetadata.callerIp AS saddr
FROM
`gcastle-gke-dev.kubecon2018.cloudaudit_googleapis_com_activity_*`,
UNNEST(protopayload_auditlog.authorizationInfo) AS authzinfo
WHERE
NOT protopayload_auditlog.requestMetadata. callerIp="127.0.0.1"
AND NOT protopayload_auditlog.requestMetadata. callerIp="::1"
AND protopayload_auditlog.requestMetadata. callerIp NOT LIKE "8.8%"
LIMIT 100
```