Open Policy Agent Gatekeeper

Security Assessment

March 10, 2020

Prepared For:
Max Smythe | *Google*
smythe@google.com

Rita Zhang | *Microsoft*
ritazh@microsoft.com

Torin Sandall | *Styra*
torin@styra.com

Chris Aniszczyk | *Cloud Native Computing Foundation*
caniszczyk@linuxfoundation.org

Prepared By:
Dominik Czarnota | *Trail of Bits*
dominik.czarnota@trailofbits.com

Claudia Richoux | *Trail of Bits*
claudia.richoux@trailofbits.com

Executive Summary

From February 18 through February 21, 2020, Cloud Native Computing Foundation (CNCF) engaged Trail of Bits to review the security of Gatekeeper. Trail of Bits conducted this assessment over the course of two person-weeks with two engineers working from commit hash 98edc61 of the Gatekeeper repository.

Gatekeeper allows enforcement of CRD-based policies over Kubernetes objects through a Kubernetes validation hook. It uses Open Policy Agent ("OPA"), a policy engine for Cloud Native environments where policies are written in the Rego policy language. It also periodically audits the existing Kubernetes objects against the specified constraints to ensure all objects continue to hold under the specified policies.

Trail of Bits performed a whitebox assessment of Open Policy Agent Gatekeeper, focusing on underlying components such as Gatekeeper controllers, auditing component, webhook, policies enforcement, and Rego query functionality. We specifically focused on the identification of problems such as improper enforcement of policies, denial of service attacks via specially-crafted policies, insufficient data validation or error handling, and more. Our assessment of Gatekeeper revealed a total of 10 findings ranging from High to Undetermined severity. Most notably, finding TOB-OPAGK-005 details an insecure configuration that allows Gatekeeper validation checks to be bypassed (e.g., by performing a denial of service attack.

In addition to manual review, Trail of Bits used Go's race detector and static analysis tools, which produced notable results for one of the OPA repositories (see Appendix C).To ensure best practices are met moving forward, we recommend using such tools within your development pipeline, as detailed in Appendix B. Additional code quality recommendations to help improve and future-proof the Gatekeeper codebase appear in Appendix D. We also reviewed the CNCF requirements criteria in Appendix F.

Project Dashboard

Application Summary

Name	Gatekeeper
Version	v3.1.0-beta.7 (git commit 98edc61)
Type	Go
Platforms	Linux

Engagement Summary

Dates	February 18-21
Method	Whitebox
Consultants Engaged	2
Level of Effort	2 person-weeks

Vulnerability Summary

Total High-Severity Issues	1	■
Total Medium-Severity Issues	0	
Total Low-Severity Issues	2	■■
Total Informational-Severity Issues	6	■■■■■■
Total Undetermined-Severity Issues	1	■
Total	10	

Category Breakdown

Configuration	3	■■■
Data Validation	1	■
Denial of Service	4	■■■■
Timing	2	■■
Total	10	

Engagement Goals

The engagement was scoped to provide a security assessment of the Open Policy Agent Gatekeeper system.

Specifically, we sought to answer the following questions:

- Is it possible to bypass the applied constraints?
- Can a policy evaluation cause a denial of service?
- Does the production deployment configuration use security best practices and minimize the attack surface?
- Is it possible to leave any used caches in an invalid state and exploit them later on?

Coverage

open-policy-agent/Gatekeeper. We performed a code review and ran Gatekeeper to check its functioning and test it against certain inputs. We also reviewed, among other elements, code paths responsible for validating and auditing constraints, metrics exporting, and various controllers' logic and cache handling. We also used static analysis tools and Go's race detector to check low-hanging fruit issues.

open-policy-agent/framework/constraints. We focused mostly on the parts directly used by Gatekeeper: the OPA client, the driver, and the template code.

open-policy-agent/opa. This dependency was reviewed with the lowest priority. We focused on implementation of the Rego policy language query evaluation partly to see if we could bypass policy checks or use it to cause denial of service attacks.

Recommendations Summary

This section aggregates all the recommendations made during the engagement. Short-term recommendations address the immediate causes of issues. Long-term recommendations pertain to the development process and long-term design goals.

Short Term

Investigate and fix data races between Gatekeeper controllers. Data races might leave data in an invalid state and can cause hard-to-spot bugs. TOB-OPAGK-001

Do not accept negative values for audit interval and constraint violations limit parameters. Validating user-provided values is important to prevent bugs. TOB-OPAGK-002, TOB-OPAGK-003

Wipe the OPA client cache before creating any controller. This will ensure proper cache cleanup if controllers are refactored. TOB-OPAGK-004

Enhance the documentation to highlight the effects of setting failurePolicy to ignore validating webhook. Additionally, log a warning if Gatekeeper is run in such a configuration or when validation requests fail. Insecure settings should be immediately obvious for users so they are aware of the possible effects. TOB-OPAGK-005

Have the client stop evaluating targets for a given webhook request after it finds the first failed target. This will prevent unnecessary load on the backend to help prevent denial of service attacks. TOB-OPAGK-006, TOB-OPAGK-009

The deployment configuration should explicitly drop Linux capabilities. This will ensure the capabilities are dropped even if other settings are changed. TOB-OPAGK-007

Implement a seccomp policy for the deployment configuration. This will limit the kernel attack surface in case an attacker gains access to Gatekeeper's container. TOB-OPAGK-008

Change the unique namespaces example to use a set instead of a list. This will prevent users from using code that doesn't fit Rego performance evaluation best practices. TOB-OPAGK-010

Long Term

Add Go's race detector into Gatekeeper's testing pipeline. This will help prevent race conditions from being introduced into future releases. TOB-OPAGK-001, Appendix E

Add unit tests for negative user-provided values and centralize validations for common parsing routines. This will help ensure the correctness of validations throughout the codebase. TOB-OPAGK-002, TOB-OPAGK-003

Add tests to ensure the shared OPA client's cache is always in its expected state. This will help prevent cache-related bugs in future versions. TOB-OPAGK-004

Change the default failurePolicy to Fail instead of Ignore in the default configuration and error out Gatekeeper if it's run with the Ignore policy. Users shouldn't be able to run an insecure configuration without explicitly specifying it. TOB-OPAGK-005

Add timeouts to queries in the OPA client and enable them in the OPA backend. Currently, timeouts are unsupported in the OPA client, and may be disabled in the OPA backend. Enabling them will reduce the impact of heavy load or denial of service attacks throughout Gatekeeper's various services. TOB-OPAGK-006

Test deployment configuration against expected mitigations. This will ensure the mitigations are in place and that Gatekeeper is functioning properly with them. TOB-OPAGK-007, TOB-OPAGK-008

Consider parallelizing the OPA client and backend, or setting up separate remote backends for separate tasks (like audits and separate webhooks). This will reduce the impact of load between services, i.e., so audits can still run during a denial of service attack targeting Rego policies through a webhook. TOB-OPAGK-009

Ensure that all examples conform to OPA's recommendations for performant policy decisions and link to them in the documentation. This will help users use more performant Rego policies and protect them against denial of service attacks. TOB-OPAGK-010

Findings Summary

#	Title	Type	Severity
1	Data races between Gatekeeper controllers	Timing	Undetermined
2	Setting audit interval to a negative value leads to an infinite audit loop	Denial of Service	Low
3	The constraintViolationsLimit can be set to a negative value	Data Validation	Informational
4	ConstraintTemplate Controller creation doesn't clear client cache	Timing	Informational
5	Default deployment uses insecure failurePolicy for validating webhook	Configuration	High
6	Policy validation doesn’t stop on first violation and doesn’t time out, potentially allowing denial of service	Denial of Service	Informational
7	The deployment configuration should explicitly drop Linux capabilities	Configuration	Informational
8	The deployment configuration does not use seccomp policies	Configuration	Low
9	OPA client and backend are shared across all parts of Gatekeeper	Denial of Service	Informational
10	Unique namespace example uses inefficient data structures	Denial of Service	Informational

1. Data races between Gatekeeper controllers

Severity: Undetermined
Type: Timing
Target: Gatekeeper controllers

Difficulty: Undetermined
Finding ID: TOB-OPAGK-001

Description

The controllers started by Gatekeeper contain a shared state that is used and mutated in a non-thread-safe manner. As a result, different controllers race between each other and may clobber the data used across them. Depending on the operations and their order, this might cause bugs if certain data is accessed in an invalid state.

The command used to launch Gatekeeper with Go's race detector along with a full execution log is included in Appendix E.

Recommendation

Short term, investigate and fix the data race cases shown in the log in Appendix E. Consider expanding the testing performed, and search for further edge cases to regressively fix.

Long term, introduce race detector tests into Gatekeeper's testing pipeline. Further, consider indexing the data dependence of each component. Before each release, ensure data race testing is performed for data sources that have multiple dependents. This will help prevent race conditions in future releases.

References

- blog.golang.org: Introducing the Go Race Detector

2. Setting audit interval to a negative value leads to an infinite audit loop

Severity: Low
Type: Denial of Service
Target: `pkg/audit/manager.go`

Difficulty: Hard
Finding ID: TOB-OPAGK-002

Description

The `--audit-interval` command line argument (Figure TOB-OPAGK-002.1) is used to set an interval time to wait either between audits (Figure TOB-OPAGK-002.2) or after constraint updates (Figure TOB-OPAGK-002.3). The flag that accepts the user's setting can be set to a negative value because it is defined as an integer. If such a value is provided, there will be no wait time between audits, which might lead to denial of service scenarios.

```
auditInterval                       = flag.Int("audit-interval", defaultAuditInterval,
"interval to run audit in seconds. defaulted to 60 secs if unspecified, 0 to disable ")
auditIntervalDeprecated             = flag.Int("auditInterval", defaultAuditInterval,
"DEPRECATED - use --audit-interval")
```

Figure TOB-OPAGK-002.1: Audit interval flags (`pkg/audit/manager.go#L41-L43`).

```
func (am *Manager) auditManagerLoop(ctx context.Context) {
        for {
                select {
                case <-ctx.Done():
                        log.Info("Audit Manager close")
                        close(am.stopper)
                        return
                default:
                        time.Sleep(time.Duration(*auditInterval) * time.Second)
                        if err := am.audit(ctx); err != nil {
                                log.Error(err, "audit manager audit() failed")
                        }
                }
        }
}
```

Figure TOB-OPAGK-002.2: The `auditManagerLoop` function which uses the `auditInterval` value to wait between audits (`pkg/audit/manager.go#L261-L275`).

```
func (am *Manager) writeAuditResults(/* (...) */) error {
        // (...)
                if len(updateConstraints) > 0 {
                        if am.ucloop != nil {
                                close(am.ucloop.stop)
                                select {
                                case <-am.ucloop.stopped:
                                case <-time.After(time.Duration(*auditInterval) * time.Second):
                                }
                        }
```

Figure TOB-OPAGK-002.3: The `writeAuditResults` function that uses `auditInterval` as a timeout for previous update constraints' action (`pkg/audit/manager.go#L384-L391`).

Exploit Scenario
Eve can control Gatekeeper’s command line flags, and sets the audit interval value to a negative number. Gatekeeper then falls into an infinite loop, producing more logs than expected and using more CPU resources than usual. Depending on the constraints and the objects present in the Kubernetes cluster, this may lead to denial of service.

Recommendation
Short term, do not accept a negative value for the audit interval argument. Data validation should ensure the provided values are within the expected range.

Long term, add unit tests to check the expected behavior of negative audit interval values. Consider centralizing validations for common parsing and validation routines into a central library to be used throughout the codebase. This will help ensure the correctness of validations throughout the codebase, and allow enforceable validation standards to be applied moving forward.

3. The constraintViolationsLimit can be set to a negative value

Severity: Informational
Type: Data Validation
Target: `pkg/audit/manager.go`

Difficulty: Hard
Finding ID: TOB-OPAGK-003

Description

The `--constraint-violations-limit` command line argument (Figure TOB-OPAGK-003.1) is used to set the "limit of number of violations per constraint." The flag can be set to a negative value because it is defined as an integer. This doesn't pose an immediate risk since the value is used in a "less than" check, so a negative value will work the same way as if it had been set to zero (Figure TOB-OPAGK-003.2). However, this behavior might introduce issues if the code is reused or refactored in the future.

```
constraintViolationsLimit           = flag.Int("constraint-violations-limit",
defaultConstraintViolationsLimit, "limit of number of violations per constraint. defaulted
to 20 violations if unspecified ")
constraintViolationsLimitDeprecated = flag.Int("constraintViolationsLimit",
defaultConstraintViolationsLimit, "DEPRECATED - use --constraint-violations-limit")
```

Figure TOB-OPAGK-003.1: Constraint violations limit flags (pkg/audit/manager.go#L42-L44).

```
func (ucloop *updateConstraintLoop) updateConstraintStatus(/* (...) */) error {
	// (...)
	for _, ar := range auditResults {
		// append statusViolations for this constraint until
constraintViolationsLimit has reached
		if len(statusViolations) < *constraintViolationsLimit {
			// (...)
		}
	}
```

Figure TOB-OPAGK-003.2: The `updateConstraintStatus` *function that uses the* `constraintViolationsLimit` *variable (pkg/audit/manager.go#L408-L427).*

Recommendation

Short term, change the `constraintViolationsLimit` flag type to `flag.Uint`, or validate its value to prevent it from being negative.

Long term, ensure the provided values are within the expected range and add unit tests to check against this behavior. This way, if a bug is introduced to this logic in future versions, it will be caught early on.

4. ConstraintTemplate Controller creation doesn't clear client cache

Severity: Informational
Type: Timing
Target: `Config and ConstraintTemplate controllers`

Difficulty: Hard
Finding ID: TOB-OPAGK-004

Description

When the config controller is created in its `Add` function (Figure TOB-OPAGK-004.1), it also wipes the OPA client cache because it might be stored remotely. However, when the constraint template controller is created, it does not wipe the client cache in its `Add` function (Figure TOB-OPAGK-004.2).

Currently, this is not a bug. Both controllers are created at the same time, in the `AddToManager` function (Figure TOB-OPAGK-004.3). However, this could be problematic in the future. For example, controllers could be created separately or the cache wiping could be removed from one controller and not added to the other.

```
// Add creates a new ConfigController and adds it to the Manager with default RBAC. The
Manager will set fields on the Controller
// and Start it when the Manager is Started.
func (a *Adder) Add(mgr manager.Manager) error {
        r, err := newReconciler(mgr, a.Opa, a.WatchManager)
        if err != nil {
                return err
        }

        // Wipe cache on start - this is to allow for the future possibility that the OPA
cache is stored remotely
        if _, err := a.Opa.RemoveData(context.Background(), target.WipeData{}); err != nil {
                return err
        }

        return add(mgr, r)
}
```

Figure TOB-OPAGK-004.1: The config controller's `Add` *function (*[`pkg/controller/config/config_controller.go#L62-L76`]*).*

```
// Add creates a new ConstraintTemplate Controller and adds it to the Manager with default
RBAC. The Manager will set fields on the Controller
// and Start it when the Manager is Started.
func (a *Adder) Add(mgr manager.Manager) error {
        r, err := newReconciler(mgr, a.Opa, a.WatchManager)
        if err != nil {
                return err
        }
        return add(mgr, r)
}
```

Figure TOB-OPAGK-004.2: The constraint template controller's `Add` *function (*`pkg/controller/constrainttemplate/constrainttemplate_controller.go#L65-L73`*).*

```
// AddToManager adds all Controllers to the Manager
func AddToManager(m manager.Manager, client *opa.Client, wm *watch.Manager) error {
	for _, a := range Injectors {
		a.InjectOpa(client)
		a.InjectWatchManager(wm)
		if err := a.Add(m); err != nil {
			return err
		}
	}
```

Figure TOB-OPAGK-004.3: The `AddToManager` *function adds controllers which are stored in the* `Injectors` *list (pkg/controller/controller.go#L37-L45).*

Recommendation

Short term, wipe the OPA client cache in the constraint template controller. If controllers are always going to use the same client, ensure the cache wiping operation is performed before creating any controller. For example, such an operation could be added to the `AddToManager` function.

Long term, add tests to Gatekeeper to ensure that the shared OPA client's cache is always in its expected state. This will help prevent cache-related bugs in future versions.

5. Default deployment uses insecure failurePolicy for validating webhook

Severity: High
Type: Configuration
Target: `deploy/gatekeeper.yaml`

Difficulty: Low
Finding ID: TOB-OPAGK-005

Description

When Gatekeeper is set up with its default deployment configuration, its validating webhook is set to ignore errors and admit all constraint requests if Gatekeeper is down (Figure TOB-OPAGK-005.1). This means a denial of service attack allows for bypassing constraints set within the Gatekeeper system.

The information about this setting is described in the `README` (Figure TOB-OPAGK-005.2) but isn't immediately obvious and might be missed by users.

```yaml
apiVersion: admissionregistration.k8s.io/v1beta1
kind: ValidatingWebhookConfiguration
metadata:
  creationTimestamp: null
  labels:
    gatekeeper.sh/system: "yes"
  name: gatekeeper-validating-webhook-configuration
webhooks:
- clientConfig:
    caBundle: Cg==
    service:
      name: gatekeeper-webhook-service
      namespace: gatekeeper-system
      path: /v1/admit
  failurePolicy: Ignore
```

Figure TOB-OPAGK-005.1: The deployment setting that makes Kubernetes ignore the response from the Gatekeeper's validation webhook when an error occurs (`deploy/gatekeeper.yaml#L384`).

```
Currently Gatekeeper is defaulting to using Ignore for the constraint requests. This is
because the webhook server currently only has one instance, which risks downtime during
actions like upgrades. As the theoretical availability improves we will likely change the
default to Fail.
```

Figure TOB-OPAGK-005.2: The Gatekeeper README, with details about the insecure setting all the way down at the bottom (`README.md`).

Exploit Scenario

Gatekeeper doesn't respond to requests (or responds to requests spottily), because of an upgrade, denial of service attack, or other problem. With the default configuration, a request that violates constraints will be accepted by the webhook and won't be noticed until an audit occurs.

Recommendation

Short term, enhance the documentation to highlight that Gatekeeper will accept forbidden requests during its downtimes, and log a warning when Gatekeeper starts with this setting. If possible, log an informational warning if a given constraint check has been timed out.

Long term, change the default `failurePolicy` to `Fail` instead of `Ignore` in the default configuration, and, if possible, throw an error if Gatekeeper is started with the `Ignore` policy. Insecure configuration should not be the default and should be prohibited if users don't explicitly want them.

6. Policy validation doesn't stop on first violation and doesn't time out, potentially allowing denial of service

Severity: Informational
Type: Denial of Service
Target: `frameworks/constraint/pkg/client/client.go`

Difficulty: Medium
Finding ID: TOB-OPAGK-006

Description

In `client.go`, the `Review` function (Figure TOB-OPAGK-006.1) queries the OPA backend for each target in order, regardless of whether any of the previous queries have failed. Requests to the backend are also made serially and do not time out. If one query hangs or takes a long time, the entire query will take too long to respond, and the validation webhook will time out. For audits, it is probably still best to get a list of all the violations for a certain target, but the webhook might want to optimize throughput. Changing this could be important for limiting the impact of one slow Rego evaluation.

```
func (c *Client) Review(ctx context.Context, obj interface{}, opts ...QueryOpt)
(*types.Responses, error) {

  // (...)

  for name, target := range c.targets {
        handled, review, err := target.HandleReview(obj)
        // Short-circuiting question applies here as well
        if err != nil {
               errMap[name] = err
               continue
        }
        if !handled {
               continue
        }
        input := map[string]interface{}{"review": review}
        resp, err := c.backend.driver.Query(ctx, fmt.Sprintf(`hooks["%s"].violation`, name),
input, drivers.Tracing(cfg.enableTracing))
        if err != nil {
               errMap[name] = err
               continue
        }
        for _, r := range resp.Results {
               if err := target.HandleViolation(r); err != nil {
                      errMap[name] = err
                      continue TargetLoop
               }
        }
        resp.Target = name
        responses.ByTarget[name] = resp
  }
```

Figure TOB-OPAGK-006.1: The `Review` function. When a violation is found, the code keeps cycling through all the targets instead of short-circuiting (open-policy-agent/framework/constraint/pkg/client/client.go#L676-L708).

Exploit Scenario

An attacker enters a malicious input into one of the webhooks, causing Gatekeeper's OPA query to take a very long time to evaluate. This will cause an undue amount of slowdown and make it easier for the attacker to perform a denial of service attack against Gatekeeper with fewer requests.

Recommendation

Short term, add an option for clients to short-circuit this logic by breaking the loop after the first failed target. This will reduce computation time on the backend by preventing spurious evaluation of targets when the client already knows the request will fail. However, it won't reduce usability for clusters when this functionality is unnecessary due to low load.

Long term, consider parallelizing the OPA client and backend, and adding timeouts to requests to the backend. The local backend driver in the OPA client does not implement timeouts.

There may be timeouts set up on a remote OPA backend, and these (if they are enabled) should be properly configured to consider the expected load and how long Kubernetes waits on the webhook before timing out. The remote backend driver uses the default options on Golang's HTTP Client, which do not include timeout, so it should be configured to time out after a few seconds, depending on the Kubernetes webhook timeout settings.

7. The deployment configuration should explicitly drop Linux capabilities

Severity: Informational
Type: Configuration
Target: `deploy/gatekeeper.yaml`

Difficulty: Hard
Finding ID: TOB-OPAGK-007

Description

The default deployment configuration does not explicitly drop Linux capabilities in its `securityContext` (Figure TOB-OPAGK-007.1). While `current` settings (`allowPrivilegeEscalation: false` and `runAsNonRoot: true`) effectively drop all capabilities, as seen on Figure TOB-OPAGK-007.2, it might be beneficial to explicitly drop Linux capabilities in cases when the underlying container engine would not respect those settings properly.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
    gatekeeper.sh/system: "yes"
  name: gatekeeper-controller-manager
  namespace: gatekeeper-system
spec:
  # (...)
  template:
    # (...)
    spec:
      containers:
        # (...)
        resources:
          securityContext:
            allowPrivilegeEscalation: false
            runAsGroup: 999
            runAsNonRoot: true
            runAsUser: 1000
```

Figure TOB-OPAGK-007.1: The `securityContext` of the Gatekeeper controller deployment (deploy/gatekeeper.yaml#L287-L357).

```
# ps auxf | grep gatekeeper
docker    9859 14.3  7.7 1331660 154000 ?      Ssl  16:55   1:16 |       \_ /manager
--audit-interval=30 --port=8443 --logtostderr --exempt-namespace=gatekeeper-system

# cat /proc/9859/status
Name:   manager
Umask:  0022
State:  S (sleeping)
Tgid:   9859
Ngid:   0
Pid:    9859
PPid:   9842
TracerPid:      0
```

```
Uid:    1000    1000    1000    1000
Gid:    999     999     999     999
# (...)
CapInh: 00000000a80425fb
CapPrm: 0000000000000000
CapEff: 0000000000000000
CapBnd: 00000000a80425fb
CapAmb: 0000000000000000
NoNewPrivs:     1
Seccomp:        0
# (...)
```

Figure TOB-OPAGK-007.2: Status of the Gatekeeper's manager process.

Recommendation

Short term, explicitly drop all capabilities in the deployment configuration. This can be done by including `drop: all` under the `securityContext` settings.

Long term, ensure the deployment configuration have all expected mitigations enabled by testing them appropriately. For example, the Linux capabilities or the `noNewPrivs` flag can be tested by checking the `/proc/PID/status` file of the Gatekeeper process.

8. The deployment configuration does not use seccomp policies

Severity: Low
Type: Configuration
Target: `deploy/gatekeeper.yaml`

Difficulty: Hard
Finding ID: TOB-OPAGK-008

Description

The default deployment configuration does not set a seccomp policy in its `securityContext` (Figure TOB-OPAGK-008.1). A seccomp policy reduces kernel attack surface by limiting the available system calls that can be run by a given process. It would effectively mitigate actions and potential privilege escalation from an attacker running arbitrary code under the Gatekeeper process or container.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
    gatekeeper.sh/system: "yes"
  name: gatekeeper-controller-manager
  namespace: gatekeeper-system
spec:
  # (...)
  template:
    # (...)
    spec:
      containers:
        # (...)
        resources:
          securityContext:
            allowPrivilegeEscalation: false
            runAsGroup: 999
            runAsNonRoot: true
            runAsUser: 1000
```

Figure TOB-OPAGK-008.1: The `securityContext` of the Gatekeeper controller deployment configuration ([`deploy/gatekeeper.yaml#L287-L357`](deploy/gatekeeper.yaml#L287-L357)).

```
# ps auxf | grep gatekeeper
docker    9859 14.3  7.7 1331660 154000 ?      Ssl  16:55   1:16  |       \_ /manager
--audit-interval=30 --port=8443 --logtostderr --exempt-namespace=gatekeeper-system

# cat /proc/9859/status | grep Seccomp
Seccomp:       0
```

Figure TOB-OPAGK-008.2: Status of the Gatekeeper's manager process showing that seccomp is not used.

Exploit Scenario

Eve gets access to Gatekeeper's container and uses a kernel exploit to escalate their privileges and take control over the entire machine.

Recommendation

Short term, implement a seccomp policy for Gatekeeper and add it to the deployment configuration. This could be based upon the default seccomp policy used by Docker and further adjusted for Gatekeeper needs.

Long term, add appropriate testing to ensure the applied seccomp policy doesn't block any new features added to Gatekeeper in the future. Consider adding documentation for developers regarding the maintenance and testing of newly added seccomp policy rules.

References

- Kubernetes.io: Configure a Security Context for a Pod or Container
- Kubernetes.io: Pod Security Policy: seccomp

9. OPA client and backend are shared across all parts of Gatekeeper

Severity: Informational	Difficulty: Medium
Type: Denial of Service	Finding ID: TOB-OPAGK-009
Target: `main.go`	

Description

The OPA client and backend are shared with both webhooks and the audit functionality. This means that if an attacker is able to slow down the backend using requests on one webhook, audits could be slowed or disabled, and the other webhook would also be disabled. As a result a denial of service attack could have a wider impact and disable more functionality than the part of Gatekeeper that was originally targeted. It was not immediately clear how multithreading or concurrency impacts the local backend and client. With a remote backend, it could be good to set up separate backends to limit impact, especially if users configure the system to ignore errors for one webhook but not the other, or if the Kubernetes API is exposed publicly.

```
backend, err := opa.NewBackend(opa.Driver(driver))
if err != nil {
	setupLog.Error(err, "unable to set up OPA backend")
	os.Exit(1)
}
client, err := backend.NewClient(opa.Targets(&target.K8sValidationTarget{}))
if err != nil {
	setupLog.Error(err, "unable to set up OPA client")
}
```

Figure TOB-OPAGK-008.1: One client and backend is initialized and used for both webhooks and audits (`main.go#L100`).

Recommendation

Short term, use the recommendations from finding TOB-OPAGK-006 to make it harder to mount denial of service attacks against the OPA backend.

Long term, consider adding a way to configure multiple backends, and document instances in which this might be useful. Depending on 1) client/backend timeout settings, 2) multithreading as discussed in the recommendations from TOB-OPAGK-006, and 3) configuration and load, this may or may not be necessary.

During the final meeting for this audit, Trail of Bits discussed another solution with the Gatekeeper team. The current scalability plan involves moving the audit process to a separate pod, adding a pool of Gatekeeper pods backing the webhooks with load balancing, and setting both webhooks to fail instead of ignoring errors. These steps will sufficiently address this finding.

10. Unique namespace example uses inefficient data structures

Severity: Informational
Difficulty: Medium
Type: Denial of Service
Finding ID: TOB-OPAGK-010
Target: `demo/basic/templates/k8suniquelabel_template.yaml`

Description

Examples in the demos and the README demonstrate how to write some common patterns for users, like ensuring uniqueness of a set of objects. To compute this, the examples use techniques with unnecessarily high computational complexity. They also often do not conform to OPA's recommended practices for efficient Rego evaluations. Slow Rego evaluation can make Gatekeeper easier to overload, and since many users are likely to copy these examples verbatim into their own constraints, such inefficient examples in the documentation could make user policies vulnerable.

```
 violation[{"msg": msg, "details": {"value": val, "label": label}}] {
    label := input.parameters.label
    val := input.review.object.metadata.labels[label]
    cluster_objs := [o | o = data.inventory.cluster[_][_][_]; not identical_cluster(o,
input.review)]
    ns_objs := [o | o = data.inventory.namespace[_][_][_][_]; not identical_namespace(o,
input.review)]
    all_objs := array.concat(cluster_objs, ns_objs)
    all_values := {val | obj = all_objs[_]; val = obj.metadata.labels[label]}
    count({val} - all_values) == 0
    msg := sprintf("label %v has duplicate value %v", [label, val])
 }
```

Figure TOB-OPAGK-010.1: This example cycles through the `cluster_objs` *and* `ns_objs` *lists (for a complexity of O(n) where n is the number of clusters or namespaces) instead of using a set (which would be O(1)) (*[`demo/basic/templates/k8suniquelabel_template.yaml#L49-L51`]*).*

Exploit Scenario

An attacker tries to mount a denial of service attack by inundating the cluster with requests while there are already 50 namespaces in the cluster. Each request does 50 string comparisons instead of one and takes much longer, so the attacker is able to overload the cluster with significantly fewer requests.

Recommendation

Short term, change the "uniqueness" examples to check for membership in a set instead of cycling through a list. If there is a way to pre-compute and store relevant strings and sets of namespace information (instead of having them generated per request), do so and document it alongside the examples so that users will do the same.

Long term, ensure that all examples conform to OPA's recommendations for performant policy decisions and link to these recommendations in the portions of documentation that discuss writing Rego for constraints.

A. Vulnerability Classifications

Vulnerability Classes	
Class	**Description**
Access Controls	Related to authorization of users and assessment of rights
Auditing and Logging	Related to auditing of actions or logging of problems
Authentication	Related to the identification of users
Configuration	Related to security configurations of servers, devices or software
Cryptography	Related to protecting the privacy or integrity of data
Data Exposure	Related to unintended exposure of sensitive information
Data Validation	Related to improper reliance on the structure or values of data
Denial of Service	Related to causing system failure
Error Reporting	Related to the reporting of error conditions in a secure fashion
Patching	Related to keeping software up to date
Session Management	Related to the identification of authenticated users
Timing	Related to race conditions, locking or order of operations
Undefined Behavior	Related to undefined behavior triggered by the program

Severity Categories	
Severity	**Description**
Informational	The issue does not pose an immediate risk, but is relevant to security best practices or Defense in Depth
Undetermined	The extent of the risk was not determined during this engagement
Low	The risk is relatively small or is not a risk the customer has indicated is important
Medium	Individual user's information is at risk, exploitation would be bad for client's reputation, moderate financial impact, possible legal implications for client

High	Large numbers of users, very bad for client’s reputation, or serious legal or financial implications

Difficulty Levels	
Difficulty	**Description**
Undetermined	The difficulty of exploit was not determined during this engagement
Low	Commonly exploited, public tools exist or can be scripted that exploit this flaw
Medium	Attackers must write an exploit, or need an in-depth knowledge of a complex system
High	The attacker must have privileged insider access to the system, may need to know extremely complex technical details or must discover other weaknesses in order to exploit this issue

B. Static Analysis Recommendations

To help improve the quality of code within the Gatekeeper and OPA codebases, there are several static analysis tools available for integration in both Git pre-commit hooks and CI/CD pipelines.

- `Go-sec` is a static analysis utility that looks for a variety of problems in Go codebases. Notably, `go-sec` will identify potential stored credentials, unhandled errors, cryptographically troubling packages, and similar problems.
- `Go-vet` is a very popular static analysis utility that searches for more go-specific problems within a codebase, such as mistakes pertaining to closures, marshaling, and unsafe pointers. `Go-vet` is integrated within the go command itself, with support for other tools through the `vettool` command line flag.
- `Staticcheck` is a static analysis utility that identifies both stylistic problems and implementation problems within a Go codebase. Note: Many of the stylistic problems `staticcheck` identifies are also indicative of potential "problem areas" in a project.
- `Ineffassign` is a static analysis utility that identifies ineffectual assignments. These ineffectual assignments often identify situations in which errors go unchecked, which could lead to undefined behavior of the program due to execution in an invalid program state.
- `Errcheck` is a static analysis utility that identifies situations in which errors are not handled appropriately.

By executing these tools within the Git pre-commit hooks, code can be analyzed for potential problems before producing a commit that will be sent to a remote. This will help developers fix problems before a CI/CD pipeline detects them and requires remediation. Additionally, integrating these tools into the CI/CD pipeline will allow double-checking to ensure these problems are not introduced into the remote.

C. OPA dependency static analysis results

Trail of Bits checked Gatekeeper, open-policy-agent/opa, and open-policy-agent/frameworks/constraint repositories against static analysis tools listed in Appendix B, and reviewed the results.

Figure C.1 presents results from `staticcheck` run over the OPA repository on `71bbdc24` commit. Only this result has been attached, as it contains code quality issues that may be worth investigating, and the other tools didn't produce notable results.

```
root@e20c4777fcfd:/host/opa# staticcheck ./... | grep -v test
ast/compile.go:202:7: const compileStageMetricPrefex is unused (U1000)
ast/index.go:649:6: type triePrinter is unused (U1000)
ast/policy.go:23:5: var hashSeed1 is unused (U1000)
ast/policy.go:1322:6: type ruleSlice is unused (U1000)
cmd/eval.go:50:2: field profileTopResults is unused (U1000)
download/download.go:85:2: '_ = <-ch' can be simplified to '<-ch' (S1005)
download/download.go:175:22: func (*Downloader).logError is unused (U1000)
download/download.go:179:22: func (*Downloader).logInfo is unused (U1000)
format/format.go:800:2: unreachable case clause:
github.com/open-policy-agent/opa/ast.Statement will always match before
*github.com/open-policy-agent/opa/ast.Head (SA4020)
format/format.go:802:2: unreachable case clause:
github.com/open-policy-agent/opa/ast.Statement will always match before
*github.com/open-policy-agent/opa/ast.Expr (SA4020)
format/format.go:804:2: unreachable case clause:
github.com/open-policy-agent/opa/ast.Statement will always match before
*github.com/open-policy-agent/opa/ast.With (SA4020)
format/format.go:806:2: unreachable case clause:
github.com/open-policy-agent/opa/ast.Statement will always match before
*github.com/open-policy-agent/opa/ast.Term (SA4020)
format/format.go:963:18: func (*writer).endMultilineSeq is unused (U1000)
internal/compiler/wasm/wasm.go:384:18: this result of append is never used, except maybe in
other appends (SA4010)
internal/planner/planner.go:1065:19: func (*Planner).planNumberFloat is unused (U1000)
internal/planner/planner.go:1079:19: func (*Planner).planNumberInt is unused (U1000)
internal/planner/planner.go:1673:19: func (*Planner).planSaveLocals is unused (U1000)
internal/storage/mock/mock.go:254:2: redundant return statement (S1023)
metrics/metrics.go:201:11: should use time.Since instead of time.Now().Sub (S1012)
plugins/bundle/plugin.go:39:2: field legacyConfig is unused (U1000)
plugins/logs/plugin.go:266:2: '_ = <-ch' can be simplified to '<-ch' (S1005)
plugins/logs/plugin.go:352:2: '_ = <-ch' can be simplified to '<-ch' (S1005)
plugins/status/plugin.go:168:2: '_ = <-ch' can be simplified to '<-ch' (S1005)
repl/repl.go:65:2: only the first constant in this group has an explicit type (SA9004)
repl/repl.go:132:2: should use for {} instead of for true {} (S1006)
repl/repl.go:167:2: should use for {} instead of for true {} (S1006)
repl/repl.go:441:2: should use 'return <expr>' instead of 'if <expr> { return <bool> };
return <bool>' (S1008)
runtime/runtime.go:331:15: the channel used with signal.Notify should be buffered (SA1017)
runtime/runtime.go:412:2: should use for range instead of for { select {} } (S1000)
server/identifier/tls.go:25:37: should omit nil check; len() for nil slices is defined as
zero (S1009)
storage/inmem/index.go:91:21: func (*indices).iter is unused (U1000)
topdown/array.go:25:2: should replace loop with arrC = append(arrC, arrA...) (S1011)
```

```
topdown/array.go:29:2: should replace loop with arrC = append(arrC, arrB...) (S1011)
topdown/errors.go:97:6: func documentConflictErr is unused (U1000)
topdown/eval.go:2243:6: func plugSlice is unused (U1000)
topdown/eval.go:2278:2: should use 'return <expr>' instead of 'if <expr> { return <bool> };
return <bool>' (S1008)
topdown/internal/jwx/jwk/key_ops.go:32:2: only the first constant in this group has an
explicit type (SA9004)
topdown/tokens.go:34:2: var jwtAlgKey is unused (U1000)
topdown/trace.go:142:2: should use 'return <expr>' instead of 'if <expr> { return <bool> };
return <bool>' (S1008)
```

Figure C.1: `staticcheck` *tool results from the OPA repository.*

D. Code Quality Recommendations

The Code Quality Recommendations appendix details areas of Gatekeeper and OPA that could be improved, but do not cause problems that would impact the project's security posture. We propose these recommendations to help prevent future errors from occurring and improve the quality of future code contributions.

Duplicated Code

There was duplicated code throughout the codebase, including the function on line 91 of `pkg/target/target.go`. The `switch` cases often had duplicate code inside them, but could be reworked and simplified by using Golang's case lists.

In other places, there is duplicated code that could be replaced by adding an instance function to the type in question; also, Queries and Support reuse the same logic, but with different variable names. Adding functions to types for printing them out or printing errors in a certain standardized format, can make code shorter and more readable.

Duplicated code is problematic. For example, when refactoring code, the developer has to find every place where the code is duplicated and change it. This makes refactors much larger and more complex, more difficult to review, and more vulnerable to bugs. Problems may be caught by tests, but if not, the codebase will potentially be more vulnerable to bugs.

Writing Modular Code

The codebase contains many very large files with unclear purpose and organization. One example is `rego.go`, which is nearly 2,000 lines. Splitting this up into multiple files by functionality and adding clear documentation for types and functions that should be exported for use by other code would make it easier to read and audit the code. Making a file per type, naming it accordingly, and splitting up helpers by type for clearer functionality and patterns of inheritance would create better modularity. This also can present issues for new developers who may not understand which functions to use for which purpose.

Writing small, modular files that can be imported into other code, defining simple and well-documented interfaces with clear purposes, and splitting things into smaller files can all be very helpful for debugging, auditing, and refactoring.

Comments and Documentation

Many functions, types, and files were uncommented and undocumented. It's much easier to work with a piece of code when its purpose and usage are marked clearly, and potential antipatterns, bugs, or vulnerabilities are documented. It is also much faster to audit, use, and test code if there is documentation specifying exactly what it should do and how it interacts with other code; otherwise, a developer must spend valuable time analyzing the source to understand the intended behavior of functions that are mostly irrelevant.

Specifically, more comments would be extremely useful when the return type of a function is a tuple of things that could be nil, or are booleans with unclear meaning. A comment documenting when the boolean in this function takes on which value, and when the second argument will be nil, would be helpful for someone trying to understand this code. Although passing in `interface{}` is not ideal because it bypasses type-checking that can catch bugs, documenting possible values for the interface in both the argument and the return value can make the code much more usable.

Document risks of using weak cryptography functions

The Rego policy language exposes weak cryptography functions such as `crypto.md5` and `crypto.sha1`. While these functions may be required by end users, they are vulnerable to hash collision attacks. The documentation should describe the risks related to their use.

Clean Control Flow

Nested loops, many `if` statements without clarifying comments, and using booleans for control flow can make it difficult to analyze the actual versus intended behavior of a function, and which logic paths will be followed. This makes usage and auditing more difficult. Generally, if it is not immediately obvious what code does, it should be refactored, or at least documented.

Using the Type System

Strongly typed languages make it easier to write secure, clean code that does one well-specified thing. They also reduce common bugs that occur if the programmer forgets to handle an edge case. Generally, leaning on the type system as much as possible is a great way to prevent bugs, as the compiler catches them before the code is run.

Passing around `interface{}` and switching on the object's type instead of having, say, a union type that strictly defines what can be passed into a function, can lead to bugs with dangerous undefined behavior. Allowing a user to pass either an object or a pointer to an object in this case creates duplicate code, and in most cases causes some sort of messiness elsewhere.

E. Running Gatekeeper with Go race detector and logs

This appendix contains more details from finding TOB-OPAGK-001.

In order to run Gatekeeper with the Go race detector, a change to the `Dockerfile` file was made (Figure E.1). Figure E.2 shows the full execution log that resulted from running Gatekeeper after this change.

```
# Build
-RUN CGO_ENABLED=0 GOOS=linux GOARCH=amd64 GO111MODULE=on go build -mod vendor -a -o manager
main.go
+RUN CGO_ENABLED=1 GOOS=linux GOARCH=amd64 GO111MODULE=on go build -race -mod vendor -a -o
manager main.go
```

Figure E.1: Changes made to Gatekeeper's `Dockerfile` to run it with Go's race detector.

```
{"level":"info","ts":1582146831.070238,"logger":"setup","msg":"Setting up controller"}
{"level":"info","ts":1582146831.070722,"logger":"setup","msg":"setting up webhooks"}
{"level":"info","ts":1582146831.0708303,"logger":"controller-runtime.webhook","msg":"registering
webhook","path":"/v1/admitlabel"}
{"level":"info","ts":1582146831.0711117,"logger":"controller-runtime.webhook","msg":"registering
webhook","path":"/v1/admit"}
{"level":"info","ts":1582146831.0712292,"logger":"webhook","msg":"cert rotation is enabled"}
{"level":"info","ts":1582146831.0717368,"logger":"setup","msg":"setting up audit"}
{"level":"info","ts":1582146831.071876,"logger":"setup","msg":"setting up upgrade"}
{"level":"info","ts":1582146831.0721786,"logger":"setup","msg":"setting up metrics"}
{"level":"info","ts":1582146831.0723379,"logger":"setup","msg":"starting manager"}
{"level":"info","ts":1582146831.074436,"logger":"metrics","msg":"Starting metrics runner"}
{"level":"info","ts":1582146831.074914,"logger":"cert-rotation","msg":"starting cert rotator controller"}
{"level":"info","ts":1582146831.0753784,"logger":"controller-runtime.webhook.webhooks","msg":"starting webhook
server"}
{"level":"info","ts":1582146831.07586,"logger":"metrics","msg":"metrics","backend":"prometheus"}
{"level":"info","ts":1582146831.0769086,"logger":"controller","msg":"Starting Audit Manager","process":"audit"}
{"level":"info","ts":1582146831.0773637,"logger":"controller","msg":"Starting Upgrade Manager","metaKind":"upgrade"}
{"level":"info","ts":1582146831.0768304,"logger":"controller-runtime.certwatcher","msg":"Updated current TLS
certificate"}
{"level":"info","ts":1582146831.0778787,"logger":"controller-runtime.webhook","msg":"serving webhook
server","host":"","port":8443}
{"level":"info","ts":1582146831.0792828,"logger":"metrics","msg":"Starting server for OpenCensus Prometheus exporter"}
{"level":"info","ts":1582146831.0806096,"logger":"controller-runtime.controller","msg":"Starting
EventSource","controller":"validating-webhook-controller","source":"kind source: /, Kind="}
{"level":"info","ts":1582146831.083381,"logger":"controller-runtime.controller","msg":"Starting
EventSource","controller":"constrainttemplate-controller","source":"kind source: /, Kind="}
{"level":"info","ts":1582146831.0865495,"logger":"controller-runtime.controller","msg":"Starting
EventSource","controller":"config-controller","source":"kind source: /, Kind="}
{"level":"info","ts":1582146831.092076,"logger":"controller-runtime.certwatcher","msg":"Starting certificate watcher"}
{"level":"info","ts":1582146831.1207092,"logger":"cert-rotation","msg":"no cert refresh needed"}
{"level":"info","ts":1582146831.1862512,"logger":"controller-runtime.controller","msg":"Starting
Controller","controller":"constrainttemplate-controller"}
{"level":"info","ts":1582146831.1871533,"logger":"controller-runtime.controller","msg":"Starting
EventSource","controller":"validating-webhook-controller","source":"kind source: admissionregistration.k8s.io/v1beta1,
Kind=ValidatingWebhookConfiguration"}
{"level":"info","ts":1582146831.189207,"logger":"controller-runtime.controller","msg":"Starting
workers","controller":"constrainttemplate-controller","worker count":1}
{"level":"info","ts":1582146831.1902976,"logger":"controller-runtime.controller","msg":"Starting
Controller","controller":"config-controller"}
{"level":"info","ts":1582146831.288414,"logger":"controller-runtime.controller","msg":"Starting
Controller","controller":"validating-webhook-controller"}
{"level":"info","ts":1582146831.2887337,"logger":"controller-runtime.controller","msg":"Starting
workers","controller":"validating-webhook-controller","worker count":1}
{"level":"info","ts":1582146831.2912018,"logger":"controller-runtime.controller","msg":"Starting
workers","controller":"config-controller","worker count":1}
{"level":"info","ts":1582146831.297111,"logger":"webhook","msg":"ensuring CA cert on ValidatingWebhookConfiguration"}
{"level":"info","ts":1582146831.3198018,"logger":"webhook","msg":"ensuring CA cert on ValidatingWebhookConfiguration"}
{"level":"info","ts":1582146831.910305,"logger":"controller","msg":"resource","metaKind":"upgrade","kind":"ConstraintT
emplate","group":"templates.gatekeeper.sh","version":"v1alpha1"}
{"level":"info","ts":1582146831.9128885,"logger":"controller","msg":"resource count","metaKind":"upgrade","count":0}
{"level":"info","ts":1582146836.0751207,"logger":"watchManager","msg":"Watcher registry found changes and/or needs
restarting","started":false,"add":[],"remove":[],"change":[]}
{"level":"info","ts":1582146836.0755532,"logger":"watchManager","msg":"restarting Watch Manager","kinds":""}
{"level":"info","ts":1582146836.0756128,"logger":"watchManager","msg":"setting up watch manager"}
{"level":"info","ts":1582146836.484632,"logger":"watchManager","msg":"Calling Manager.Start()","kinds":[]}
==================
WARNING: DATA RACE
Write at 0x00c000519848 by goroutine 94:
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:102 +0x1c0
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).Handle()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/webhook.go
:135 +0xed
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).ServeHTTP()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/http.go:87
+0x10b5
  sigs.k8s.io/controller-runtime/pkg/webhook.instrumentedHook.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:117
+0x11c
  net/http.HandlerFunc.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2007 +0x51
  net/http.(*ServeMux).ServeHTTP()
      /usr/local/go/src/net/http/server.go:2387 +0x288
  net/http.serverHandler.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2802 +0xce
  net/http.(*conn).serve()
      /usr/local/go/src/net/http/server.go:1890 +0x837

Previous write at 0x00c000519848 by goroutine 29:
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:102 +0x1c0
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).Handle()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/webhook.go
:135 +0xed
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).ServeHTTP()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/http.go:87
+0x10b5
  sigs.k8s.io/controller-runtime/pkg/webhook.instrumentedHook.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:117
+0x11c
  net/http.HandlerFunc.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2007 +0x51
  net/http.(*ServeMux).ServeHTTP()
      /usr/local/go/src/net/http/server.go:2387 +0x288
  net/http.serverHandler.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2802 +0xce
  net/http.(*conn).serve()
      /usr/local/go/src/net/http/server.go:1890 +0x837

Goroutine 94 (running) created at:
  net/http.(*Server).Serve()
      /usr/local/go/src/net/http/server.go:2927 +0x5be
  sigs.k8s.io/controller-runtime/pkg/webhook.(*Server).Start()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:189
+0xcc2
  sigs.k8s.io/controller-runtime/pkg/manager.(*controllerManager).startNonLeaderElectionRunnables.func1()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/manager/internal.go:455
+0x83

Goroutine 29 (running) created at:
  net/http.(*Server).Serve()
      /usr/local/go/src/net/http/server.go:2927 +0x5be
  sigs.k8s.io/controller-runtime/pkg/webhook.(*Server).Start()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:189
+0xcc2
  sigs.k8s.io/controller-runtime/pkg/manager.(*controllerManager).startNonLeaderElectionRunnables.func1()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/manager/internal.go:455
+0x83
==================
==================
WARNING: DATA RACE
Read at 0x00c00025cc70 by goroutine 94:
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*reporter).ReportRequest()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/stats_reporter.go:58 +0xce
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:141 +0xfe
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:173 +0xfb4
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).Handle()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/webhook.go
:135 +0xed
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).ServeHTTP()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/http.go:87
+0x10b5
  sigs.k8s.io/controller-runtime/pkg/webhook.instrumentedHook.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:117
+0x11c
  net/http.HandlerFunc.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2007 +0x51
  net/http.(*ServeMux).ServeHTTP()
      /usr/local/go/src/net/http/server.go:2387 +0x288
  net/http.serverHandler.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2802 +0xce
  net/http.(*conn).serve()
      /usr/local/go/src/net/http/server.go:1890 +0x837

Previous write at 0x00c00025cc70 by goroutine 29:
  [failed to restore the stack]

Goroutine 94 (running) created at:
  net/http.(*Server).Serve()
      /usr/local/go/src/net/http/server.go:2927 +0x5be
  sigs.k8s.io/controller-runtime/pkg/webhook.(*Server).Start()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:189
+0xcc2
  sigs.k8s.io/controller-runtime/pkg/manager.(*controllerManager).startNonLeaderElectionRunnables.func1()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/manager/internal.go:455
+0x83

Goroutine 29 (running) created at:
  net/http.(*Server).Serve()
      /usr/local/go/src/net/http/server.go:2927 +0x5be
  sigs.k8s.io/controller-runtime/pkg/webhook.(*Server).Start()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:189
+0xcc2
  sigs.k8s.io/controller-runtime/pkg/manager.(*controllerManager).startNonLeaderElectionRunnables.func1()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/manager/internal.go:455
+0x83
==================
==================
WARNING: DATA RACE
Read at 0x00c000bec580 by goroutine 94:
  context.(*valueCtx).Value()
      /usr/local/go/src/context/context.go:517 +0x42
  go.opencensus.io/tag.New()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/go.opencensus.io/tag/context.go:25 +0xcc
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*reporter).ReportRequest()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/stats_reporter.go:57 +0x10a
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:141 +0xfe
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:173 +0xfb4
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).Handle()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/webhook.go
:135 +0xed
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).ServeHTTP()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/http.go:87
+0x10b5
  sigs.k8s.io/controller-runtime/pkg/webhook.instrumentedHook.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:117
```

```
+0x11c
  net/http.HandlerFunc.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2007 +0x51
  net/http.(*ServeMux).ServeHTTP()
      /usr/local/go/src/net/http/server.go:2387 +0x288
  net/http.serverHandler.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2802 +0xce
  net/http.(*conn).serve()
      /usr/local/go/src/net/http/server.go:1890 +0x837

Previous write at 0x00c000bec580 by goroutine 29:
  [failed to restore the stack]

Goroutine 94 (running) created at:
  net/http.(*Server).Serve()
      /usr/local/go/src/net/http/server.go:2927 +0x5be
  sigs.k8s.io/controller-runtime/pkg/webhook.(*Server).Start()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:189
+0xcc2
  sigs.k8s.io/controller-runtime/pkg/manager.(*controllerManager).startNonLeaderElectionRunnables.func1()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/manager/internal.go:455
+0x83

Goroutine 29 (running) created at:
  net/http.(*Server).Serve()
      /usr/local/go/src/net/http/server.go:2927 +0x5be
  sigs.k8s.io/controller-runtime/pkg/webhook.(*Server).Start()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:189
+0xcc2
  sigs.k8s.io/controller-runtime/pkg/manager.(*controllerManager).startNonLeaderElectionRunnables.func1()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/manager/internal.go:455
+0x83
==================
==================
WARNING: DATA RACE
Read at 0x00c000bec590 by goroutine 94:
  context.(*valueCtx).Value()
      /usr/local/go/src/context/context.go:518 +0xfe
  go.opencensus.io/tag.New()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/go.opencensus.io/tag/context.go:25 +0xcc
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*reporter).ReportRequest()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/stats_reporter.go:57 +0x10a
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:141 +0xfe
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:173 +0xfb4
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).Handle()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/webhook.go
:135 +0xed
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).ServeHTTP()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/http.go:87
+0x10b5
  sigs.k8s.io/controller-runtime/pkg/webhook.instrumentedHook.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:117
+0x11c
  net/http.HandlerFunc.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2007 +0x51
  net/http.(*ServeMux).ServeHTTP()
      /usr/local/go/src/net/http/server.go:2387 +0x288
  net/http.serverHandler.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2802 +0xce
  net/http.(*conn).serve()
      /usr/local/go/src/net/http/server.go:1890 +0x837

Previous write at 0x00c000bec590 by goroutine 29:
  [failed to restore the stack]

Goroutine 94 (running) created at:
  net/http.(*Server).Serve()
      /usr/local/go/src/net/http/server.go:2927 +0x5be
  sigs.k8s.io/controller-runtime/pkg/webhook.(*Server).Start()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:189
+0xcc2
```

```
  sigs.k8s.io/controller-runtime/pkg/manager.(*controllerManager).startNonLeaderElectionRunnables.func1()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/manager/internal.go:455
+0x83

Goroutine 29 (running) created at:
  net/http.(*Server).Serve()
      /usr/local/go/src/net/http/server.go:2927 +0x5be
  sigs.k8s.io/controller-runtime/pkg/webhook.(*Server).Start()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:189
+0xcc2
  sigs.k8s.io/controller-runtime/pkg/manager.(*controllerManager).startNonLeaderElectionRunnables.func1()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/manager/internal.go:455
+0x83
==================
==================
WARNING: DATA RACE
Read at 0x00c00085d430 by goroutine 94:
  go.opencensus.io/tag.New()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/go.opencensus.io/tag/map.go:191 +0x106
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*reporter).ReportRequest()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/stats_reporter.go:57 +0x10a
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:141 +0xfe
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:173 +0xfb4
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).Handle()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/webhook.go
:135 +0xed
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).ServeHTTP()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/http.go:87
+0x10b5
  sigs.k8s.io/controller-runtime/pkg/webhook.instrumentedHook.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:117
+0x11c
  net/http.HandlerFunc.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2007 +0x51
  net/http.(*ServeMux).ServeHTTP()
      /usr/local/go/src/net/http/server.go:2387 +0x288
  net/http.serverHandler.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2802 +0xce
  net/http.(*conn).serve()
      /usr/local/go/src/net/http/server.go:1890 +0x837

Previous write at 0x00c00085d430 by goroutine 29:
  [failed to restore the stack]

Goroutine 94 (running) created at:
  net/http.(*Server).Serve()
      /usr/local/go/src/net/http/server.go:2927 +0x5be
  sigs.k8s.io/controller-runtime/pkg/webhook.(*Server).Start()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:189
+0xcc2
  sigs.k8s.io/controller-runtime/pkg/manager.(*controllerManager).startNonLeaderElectionRunnables.func1()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/manager/internal.go:455
+0x83

Goroutine 29 (running) created at:
  net/http.(*Server).Serve()
      /usr/local/go/src/net/http/server.go:2927 +0x5be
  sigs.k8s.io/controller-runtime/pkg/webhook.(*Server).Start()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:189
+0xcc2
  sigs.k8s.io/controller-runtime/pkg/manager.(*controllerManager).startNonLeaderElectionRunnables.func1()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/manager/internal.go:455
+0x83
==================
==================
WARNING: DATA RACE
Read at 0x00c0009c0050 by goroutine 94:
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*reporter).ReportRequest()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/stats_reporter.go:58 +0xce
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:141 +0xfe
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:173 +0xfb4
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).Handle()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/webhook.go
:135 +0xed
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).ServeHTTP()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/http.go:87
+0x10b5
  sigs.k8s.io/controller-runtime/pkg/webhook.instrumentedHook.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:117
+0x11c
  net/http.HandlerFunc.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2007 +0x51
  net/http.(*ServeMux).ServeHTTP()
      /usr/local/go/src/net/http/server.go:2387 +0x288
  net/http.serverHandler.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2802 +0xce
  net/http.(*conn).serve()
      /usr/local/go/src/net/http/server.go:1890 +0x837

Previous write at 0x00c0009c0050 by goroutine 29:
  github.com/open-policy-agent/gatekeeper/pkg/webhook.newStatsReporter()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/stats_reporter.go:52 +0xb6
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:98 +0x17d
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).Handle()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/webhook.go
:135 +0xed
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).ServeHTTP()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/http.go:87
+0x10b5
  sigs.k8s.io/controller-runtime/pkg/webhook.instrumentedHook.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:117
+0x11c
  net/http.HandlerFunc.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2007 +0x51
  net/http.(*ServeMux).ServeHTTP()
      /usr/local/go/src/net/http/server.go:2387 +0x288
  net/http.serverHandler.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2802 +0xce
  net/http.(*conn).serve()
      /usr/local/go/src/net/http/server.go:1890 +0x837

Goroutine 94 (running) created at:
  net/http.(*Server).Serve()
      /usr/local/go/src/net/http/server.go:2927 +0x5be
  sigs.k8s.io/controller-runtime/pkg/webhook.(*Server).Start()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:189
+0xcc2
  sigs.k8s.io/controller-runtime/pkg/manager.(*controllerManager).startNonLeaderElectionRunnables.func1()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/manager/internal.go:455
+0x83

Goroutine 29 (running) created at:
  net/http.(*Server).Serve()
      /usr/local/go/src/net/http/server.go:2927 +0x5be
  sigs.k8s.io/controller-runtime/pkg/webhook.(*Server).Start()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:189
+0xcc2
  sigs.k8s.io/controller-runtime/pkg/manager.(*controllerManager).startNonLeaderElectionRunnables.func1()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/manager/internal.go:455
+0x83
==================
==================
WARNING: DATA RACE
Read at 0x00c000b92340 by goroutine 94:
  context.(*valueCtx).Value()
      /usr/local/go/src/context/context.go:517 +0x42
  go.opencensus.io/tag.New()
```

```
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/go.opencensus.io/tag/context.go:25 +0xcc
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*reporter).ReportRequest()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/stats_reporter.go:57 +0x10a
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:141 +0xfe
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:173 +0xfb4
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).Handle()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/webhook.go
:135 +0xed
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).ServeHTTP()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/http.go:87
+0x10b5
  sigs.k8s.io/controller-runtime/pkg/webhook.instrumentedHook.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:117
+0x11c
  net/http.HandlerFunc.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2007 +0x51
  net/http.(*ServeMux).ServeHTTP()
      /usr/local/go/src/net/http/server.go:2387 +0x288
  net/http.serverHandler.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2802 +0xce
  net/http.(*conn).serve()
      /usr/local/go/src/net/http/server.go:1890 +0x837

Previous write at 0x00c000b92340 by goroutine 29:
  context.WithValue()
      /usr/local/go/src/context/context.go:487 +0xc8
  go.opencensus.io/tag.New()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/go.opencensus.io/tag/context.go:38 +0x806
  github.com/open-policy-agent/gatekeeper/pkg/webhook.newStatsReporter()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/stats_reporter.go:45 +0x6c
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:98 +0x17d
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).Handle()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/webhook.go
:135 +0xed
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).ServeHTTP()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/http.go:87
+0x10b5
  sigs.k8s.io/controller-runtime/pkg/webhook.instrumentedHook.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:117
+0x11c
  net/http.HandlerFunc.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2007 +0x51
  net/http.(*ServeMux).ServeHTTP()
      /usr/local/go/src/net/http/server.go:2387 +0x288
  net/http.serverHandler.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2802 +0xce
  net/http.(*conn).serve()
      /usr/local/go/src/net/http/server.go:1890 +0x837

Goroutine 94 (running) created at:
  net/http.(*Server).Serve()
      /usr/local/go/src/net/http/server.go:2927 +0x5be
  sigs.k8s.io/controller-runtime/pkg/webhook.(*Server).Start()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:189
+0xcc2
  sigs.k8s.io/controller-runtime/pkg/manager.(*controllerManager).startNonLeaderElectionRunnables.func1()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/manager/internal.go:455
+0x83

Goroutine 29 (running) created at:
  net/http.(*Server).Serve()
      /usr/local/go/src/net/http/server.go:2927 +0x5be
  sigs.k8s.io/controller-runtime/pkg/webhook.(*Server).Start()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:189
+0xcc2
  sigs.k8s.io/controller-runtime/pkg/manager.(*controllerManager).startNonLeaderElectionRunnables.func1()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/manager/internal.go:455
+0x83
```

```
==================
==================
WARNING: DATA RACE
Read at 0x00c000b92350 by goroutine 94:
  context.(*valueCtx).Value()
      /usr/local/go/src/context/context.go:518 +0xfe
  go.opencensus.io/tag.New()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/go.opencensus.io/tag/context.go:25 +0xcc
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*reporter).ReportRequest()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/stats_reporter.go:57 +0x10a
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:141 +0xfe
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:173 +0xfb4
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).Handle()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/webhook.go
:135 +0xed
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).ServeHTTP()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/http.go:87
+0x10b5
  sigs.k8s.io/controller-runtime/pkg/webhook.instrumentedHook.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:117
+0x11c
  net/http.HandlerFunc.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2007 +0x51
  net/http.(*ServeMux).ServeHTTP()
      /usr/local/go/src/net/http/server.go:2387 +0x288
  net/http.serverHandler.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2802 +0xce
  net/http.(*conn).serve()
      /usr/local/go/src/net/http/server.go:1890 +0x837

Previous write at 0x00c000b92350 by goroutine 29:
  context.WithValue()
      /usr/local/go/src/context/context.go:487 +0xf5
  go.opencensus.io/tag.New()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/go.opencensus.io/tag/context.go:38 +0x806
  github.com/open-policy-agent/gatekeeper/pkg/webhook.newStatsReporter()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/stats_reporter.go:45 +0x6c
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:98 +0x17d
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).Handle()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/webhook.go
:135 +0xed
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).ServeHTTP()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/http.go:87
+0x10b5
  sigs.k8s.io/controller-runtime/pkg/webhook.instrumentedHook.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:117
+0x11c
  net/http.HandlerFunc.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2007 +0x51
  net/http.(*ServeMux).ServeHTTP()
      /usr/local/go/src/net/http/server.go:2387 +0x288
  net/http.serverHandler.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2802 +0xce
  net/http.(*conn).serve()
      /usr/local/go/src/net/http/server.go:1890 +0x837

Goroutine 94 (running) created at:
  net/http.(*Server).Serve()
      /usr/local/go/src/net/http/server.go:2927 +0x5be
  sigs.k8s.io/controller-runtime/pkg/webhook.(*Server).Start()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:189
+0xcc2
  sigs.k8s.io/controller-runtime/pkg/manager.(*controllerManager).startNonLeaderElectionRunnables.func1()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/manager/internal.go:455
+0x83

Goroutine 29 (running) created at:
  net/http.(*Server).Serve()
      /usr/local/go/src/net/http/server.go:2927 +0x5be
  sigs.k8s.io/controller-runtime/pkg/webhook.(*Server).Start()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:189
+0xcc2
  sigs.k8s.io/controller-runtime/pkg/manager.(*controllerManager).startNonLeaderElectionRunnables.func1()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/manager/internal.go:455
+0x83
==================
==================
WARNING: DATA RACE
Read at 0x00c000646038 by goroutine 94:
  go.opencensus.io/tag.New()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/go.opencensus.io/tag/map.go:191 +0x106
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*reporter).ReportRequest()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/stats_reporter.go:57 +0x10a
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:141 +0xfe
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:173 +0xfb4
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).Handle()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/webhook.go
:135 +0xed
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).ServeHTTP()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/http.go:87
+0x10b5
  sigs.k8s.io/controller-runtime/pkg/webhook.instrumentedHook.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:117
+0x11c
  net/http.HandlerFunc.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2007 +0x51
  net/http.(*ServeMux).ServeHTTP()
      /usr/local/go/src/net/http/server.go:2387 +0x288
  net/http.serverHandler.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2802 +0xce
  net/http.(*conn).serve()
      /usr/local/go/src/net/http/server.go:1890 +0x837

Previous write at 0x00c000646038 by goroutine 29:
  go.opencensus.io/tag.New()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/go.opencensus.io/tag/map.go:92 +0x7a
  github.com/open-policy-agent/gatekeeper/pkg/webhook.newStatsReporter()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/stats_reporter.go:45 +0x6c
  github.com/open-policy-agent/gatekeeper/pkg/webhook.(*validationHandler).Handle()
      /go/src/github.com/open-policy-agent/gatekeeper/pkg/webhook/policy.go:98 +0x17d
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).Handle()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/webhook.go
:135 +0xed
  sigs.k8s.io/controller-runtime/pkg/webhook/admission.(*Webhook).ServeHTTP()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/admission/http.go:87
+0x10b5
  sigs.k8s.io/controller-runtime/pkg/webhook.instrumentedHook.func1()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:117
+0x11c
  net/http.HandlerFunc.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2007 +0x51
  net/http.(*ServeMux).ServeHTTP()
      /usr/local/go/src/net/http/server.go:2387 +0x288
  net/http.serverHandler.ServeHTTP()
      /usr/local/go/src/net/http/server.go:2802 +0xce
  net/http.(*conn).serve()
      /usr/local/go/src/net/http/server.go:1890 +0x837

Goroutine 94 (running) created at:
  net/http.(*Server).Serve()
      /usr/local/go/src/net/http/server.go:2927 +0x5be
  sigs.k8s.io/controller-runtime/pkg/webhook.(*Server).Start()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:189
+0xcc2
  sigs.k8s.io/controller-runtime/pkg/manager.(*controllerManager).startNonLeaderElectionRunnables.func1()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/manager/internal.go:455
+0x83
```

```
Goroutine 29 (running) created at:
  net/http.(*Server).Serve()
      /usr/local/go/src/net/http/server.go:2927 +0x5be
  sigs.k8s.io/controller-runtime/pkg/webhook.(*Server).Start()
      /go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/webhook/server.go:189
+0xcc2
  sigs.k8s.io/controller-runtime/pkg/manager.(*controllerManager).startNonLeaderElectionRunnables.func1()

/go/src/github.com/open-policy-agent/gatekeeper/vendor/sigs.k8s.io/controller-runtime/pkg/manager/internal.go:455
+0x83
==================
{"level":"info","ts":1582146861.0783536,"logger":"controller","msg":"auditing constraints and
violations","process":"audit","audit_id":"2020-02-19T21:14:21Z","event_type":"audit_started"}
{"level":"info","ts":1582146861.5009127,"logger":"controller","msg":"Auditing via discovery
client","process":"audit","audit_id":"2020-02-19T21:14:21Z"}
{"level":"info","ts":1582146874.199,"logger":"controller","msg":"Audit discovery client
results","process":"audit","audit_id":"2020-02-19T21:14:21Z","violations":0}
{"level":"info","ts":1582146874.201328,"logger":"controller","msg":"no constraint is found with
apiversion","process":"audit","audit_id":"2020-02-19T21:14:21Z","constraint
apiversion":"constraints.gatekeeper.sh/v1beta1"}
{"level":"info","ts":1582146874.2015147,"logger":"controller","msg":"auditing is
complete","process":"audit","audit_id":"2020-02-19T21:14:21Z","event_type":"audit_finished"}
```

Figure E.2: Full logs from running Gatekeeper with Go's race detector enabled.

F. CNCF Requirements Criteria Review

This appendix lists general improvements based upon best practices for Free/Libre and Open Source Software (FLOSS) projects that could be applied to the OPA Gatekeeper project. Introducing those changes should help accommodate the CNCF project graduation criteria.

Detail the contribution process in repository's README. While there is a "Want to help?" section, it doesn't explain the contribution process.

Use Github issues and pull request templates. This will make it easier to keep a consistent format for submitted issues, feature requests, and pull requests.

Define how to report security bugs. This has been mentioned in issue #264 but is yet to be added.

Document risks of using weak cryptography functions. This has been described in the Code Quality Recommendations.

Add more tests and track test coverage in pull requests. Increase project test coverage since it's currently relatively low (Figure F.1). Introduce a component to the project's CI system to track changes in code coverage as the project matures.

Add static analysis tools for the OPA project. Several notable tools have been described in Appendix B: Static Analysis Recommendations.

Use Go's race detector. As described in TOB-OPAGK-001 and Appendix E, using Go's race detector would help to prevent race conditions that may lead to security vulnerabilities.

```
GO111MODULE=on go test -mod vendor ./pkg/... -coverprofile cover.out
ok       github.com/open-policy-agent/gatekeeper/pkg/audit       0.111s    coverage: 5.1% of statements
?        github.com/open-policy-agent/gatekeeper/pkg/controller  [no test files]
ok       github.com/open-policy-agent/gatekeeper/pkg/controller/config     11.535s   coverage: 56.0% of statements
ok       github.com/open-policy-agent/gatekeeper/pkg/controller/constraint 0.148s    coverage: 13.3% of statements
ok       github.com/open-policy-agent/gatekeeper/pkg/controller/constrainttemplate  19.747s   coverage: 54.1% of
statements
?        github.com/open-policy-agent/gatekeeper/pkg/controller/sync      [no test files]
?        github.com/open-policy-agent/gatekeeper/pkg/logging     [no test files]
ok       github.com/open-policy-agent/gatekeeper/pkg/metrics     0.142s    coverage: 25.0% of statements
ok       github.com/open-policy-agent/gatekeeper/pkg/target      9.019s    coverage: 67.9% of statements
?        github.com/open-policy-agent/gatekeeper/pkg/upgrade     [no test files]
ok       github.com/open-policy-agent/gatekeeper/pkg/util        0.014s    coverage: 60.5% of statements
ok       github.com/open-policy-agent/gatekeeper/pkg/util/constraint      0.028s    coverage: 72.2% of statements
ok       github.com/open-policy-agent/gatekeeper/pkg/watch       0.090s    coverage: 69.9% of statements
ok       github.com/open-policy-agent/gatekeeper/pkg/webhook     7.084s    coverage: 40.7% of statements
```

Figure F.1: Gatekeeper's test coverage, generated by executing `make native-test` *in the project's directory.*