Commit

✓ fix: AdmissionPolicyGroup must not have context aware resources

Browse files

```
Prior to this commit, it was possible to declare AdmissionPolicyGroup
resources with policies that had context aware resources.

Context awareness allows a policy to query the Kubernetes API with `list`
and `get` verbs. The queries are run using the ServiceAccount of the
Policy Server instance that hosts the policy.

AdmissionPolicyGroup are namespaced CRDs which can be created by
non-admin Kubernetes users.
The ability to deploy AdmissionPolicyGroup with context aware resources
exposes the cluster to potential leaks of data.
For example, a non-admin user, might create an AdmissionPolicyGroup with
one of its policies that has access to Kubernetes Secrets. The policy
would then be able to `get` Secrets that are not available to the
non-admin user (if the Policy Server is running with a ServiceAccount
that has been explicitly configured to have `get` access to all the
Secrets of the cluster).

This commit fixes the by removing the `ContextAwareResources` field
from the `AdmissionPolicyGroup`.

== What happens after this commit is deployed

The existing AdmissionPolicyGroup will be automatically updated and they
will lose their `.spec.policies.[*].ContextAwareResources` field (if set).

The kubewarden controller will also automatically reconcile the Policy
Server configuration.

Signed-off-by: Flavio Castelli <fcastelli@suse.com>
```

main (#984)
v1.21.0 v1.21.0-rc2

flavio committed 2 weeks ago Verified 1 parent 04225f4 commit 51a88df

Showing **13 changed files** with **522 additions** and **237 deletions**.

Whitesp... Ignore whitesp... S... Uni...

Filter changed files

19

api/policies/v1/admissionpolicygroup_types.go

```
75  75  }
76  76
77  77  func (r *AdmissionPolicyGroup) IsContextAware() bool {
78      -       for _, policy := range r.Spec.Policies {
79      -               if len(policy.ContextAwareResources) > 0 {
80      -                       return true
81      -               }
82      -       }
    78  +       // We return false here because AdmissionPolicyGroup have no access to context aware resources.
83  79          return false
84  80  }
85  81
91  87          return ""
92  88  }
93  89
94      - func (r *AdmissionPolicyGroup) GetPolicyGroupMembers() PolicyGroupMembers {
95      -       return r.Spec.Policies
    90  + func (r *AdmissionPolicyGroup) GetPolicyGroupMembersWithContext() PolicyGroupMembersWithContext {
    91  +       policies := make(PolicyGroupMembersWithContext)
    92  +       for name, policy := range r.Spec.Policies {
    93  +               policies[name] = PolicyGroupMemberWithContext{
    94  +                       PolicyGroupMember: policy,
    95  +                       // AdmissionPolicyGroup does not have access to context aware resources.
    96  +                       ContextAwareResources: []ContextAwareResource{},
    97  +               }
    98  +       }
    99  +
   100  +       return policies
96  101 }
97  102
98  103 func (r *AdmissionPolicyGroup) GetSettings() runtime.RawExtension {
```

14

api/policies/v1/clusteradmissionpolicygroup_types.go

```
22  22          "k8s.io/apimachinery/pkg/runtime"
23  23  )
24  24
    25  + type ClusterPolicyGroupSpec struct {
    26  +       GroupSpec `json:""`
```

```
                27 +
                28 +         // Policies is a list of policies that are part of the group that will
                29 +         // be available to be called in the evaluation expression field.
                30 +         // Each policy in the group should be a Kubewarden policy.
                31 +         // +kubebuilder:validation:Required
                32 +         Policies PolicyGroupMembersWithContext `json:"policies"`
                33 + }
                34 +
25              35   // ClusterAdmissionPolicyGroupSpec defines the desired state of ClusterAdmissionPolicyGroup.
26              36   type ClusterAdmissionPolicyGroupSpec struct {
27                 - PolicyGroupSpec `json:""`
                37 + ClusterPolicyGroupSpec `json:""`
28              38
29              39           // NamespaceSelector decides whether to run the webhook on an object based
30              40           // on whether the namespace for that object matches the selector. If the
148             158          return &r.Status
149             159  }
150             160
151                - func (r *ClusterAdmissionPolicyGroup) GetPolicyGroupMembers() PolicyGroupMembers {
                161 + func (r *ClusterAdmissionPolicyGroup) GetPolicyGroupMembersWithContext() PolicyGroupMembersWithContext {
152             162          return r.Spec.Policies
153             163  }
154             164
```

44 api/policies/v1/factories.go

```
314             314                  },
315             315                  Spec: AdmissionPolicyGroupSpec{
316             316                          PolicyGroupSpec: PolicyGroupSpec{
317                -                                 PolicyServer: f.policyServer,
318                -                                 Policies: f.policyMembers,
319                -                                 Expression: f.expression,
320                -                                 Rules: f.rules,
321                -                                 MatchConditions: f.matchConds,
322                -                                 Mode: f.mode,
```

```
                                GroupSpec:
GroupSpec{
+
PolicyServer:    f.policyServer,
+
Expression:      f.expression,
+                                       Rules:
f.rules,
+
MatchConditions: f.matchConds,
+                                       Mode:
f.mode,
+                                },
+                                Policies:
f.policyMembers,
                        },
                },
        }
        policyServer  string
        rules
[]admissionregistrationv1.RuleWithOperations
        expression    string
-       policyMembers PolicyGroupMembers
+       policyMembers
PolicyGroupMembersWithContext
        matchConds
[]admissionregistrationv1.MatchCondition
        mode          PolicyMode
}
                        },
                },
                expression: "pod_privileged() &&
user_group_psp()",
-               policyMembers: PolicyGroupMembers{
+               policyMembers:
PolicyGroupMembersWithContext{
                        "pod_privileged": {
-                               Module:
"registry://ghcr.io/kubewarden/tests/pod-
privileged:v0.2.5",
+                               PolicyGroupMember:
PolicyGroupMember{
+                                       Module:
"registry://ghcr.io/kubewarden/tests/pod-
privileged:v0.2.5",
+                               },
                        },
                        "user_group_psp": {
-                               Module:
"registry://ghcr.io/kubewarden/tests/user-group-
psp:v0.4.9",
```

```
+                                PolicyGroupMember: PolicyGroupMember{
+                                        Module: "registry://ghcr.io/kubewarden/tests/user-group-psp:v0.4.9",
+                                },
                        },
                },
                matchConds: []admissionregistrationv1.MatchCondition{
        return f
}

- func (f *ClusterAdmissionPolicyGroupFactory) WithMembers(members PolicyGroupMembers) *ClusterAdmissionPolicyGroupFactory {
+ func (f *ClusterAdmissionPolicyGroupFactory) WithMembers(members PolicyGroupMembersWithContext) *ClusterAdmissionPolicyGroupFactory {
        f.policyMembers = members
        return f
}
                        },
                },
                Spec: ClusterAdmissionPolicyGroupSpec{
-                       PolicyGroupSpec: PolicyGroupSpec{
-                               PolicyServer: f.policyServer,
-                               Policies: f.policyMembers,
-                               Expression: f.expression,
-                               Rules: f.rules,
-                               MatchConditions: f.matchConds,
-                               Mode: f.mode,
+                       ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
+                               GroupSpec: GroupSpec{
+                                       PolicyServer:    f.policyServer,
+                                       Expression:      f.expression,
+                                       Rules: f.rules,
+                                       MatchConditions: f.matchConds,
```

```
                                                    Mode:
f.mode,
                                    },
                                    Policies:
f.policyMembers,
                            },
                    },
            }
```

2 api/policies/v1/policy.go

```
// +kubebuilder:object:generate:=false
type PolicyGroup interface {
        Policy
-       GetPolicyGroupMembers() PolicyGroupMembers
+       GetPolicyGroupMembersWithContext()
PolicyGroupMembersWithContext
        GetExpression() string
        GetMessage() string
}
```

12 api/policies/v1/policy_types.go

```
        //
+kubebuilder:pruning:PreserveUnknownFields
        // x-kubernetes-embedded-resource: false
        Settings runtime.RawExtension
`json:"settings,omitempty"`
+ }
+
+ type PolicyGroupMembersWithContext
map[string]PolicyGroupMemberWithContext
+
+ type PolicyGroupMemberWithContext struct {
+       PolicyGroupMember `json:""`

        // List of Kubernetes resources the policy
is allowed to access at evaluation time.
        // Access to these resources is done using
the `ServiceAccount` of the PolicyServer
        ContextAwareResources
[]ContextAwareResource
`json:"contextAwareResources,omitempty"`
}

- type PolicyGroupSpec struct {
+ type GroupSpec struct {
        // PolicyServer identifies an existing
PolicyServer resource.
        // +kubebuilder:default:=default
        // +optional
        // returned in the warning field of the
response.
```

```
303 309          // +kubebuilder:validation:Required
304 310          Message string `json:"message"`
    311 + }
    312 +
    313 + type PolicyGroupSpec struct {
    314 +          GroupSpec `json:""`
305 315
306 316          // Policies is a list of policies that are part of the group that will
307 317          // be available to be called in the evaluation expression field.
```

6 api/policies/v1/policygroup_validation.go

```
56 56  // validatePolicyGroupMembers validates that a policy group has at least one policy member.
57 57  func validatePolicyGroupMembers(policyGroup PolicyGroup) field.ErrorList {
58 58          var allErrors field.ErrorList
59    - -       if len(policyGroup.GetPolicyGroupMembers()) == 0 {
   59 + +       if len(policyGroup.GetPolicyGroupMembersWithContext()) == 0 {
60 60                  allErrors = append(allErrors, field.Required(field.NewPath("spec").Child("policies"), "policy groups must have at least one policy member"))
61 61          }
62    - -       for memberName := range policyGroup.GetPolicyGroupMembers() {
   62 + +       for memberName := range policyGroup.GetPolicyGroupMembersWithContext() {
63 63                  _, matchReservedSymbol := celReservedSymbols[memberName]
64 64                  if len(memberName) == 0 || matchReservedSymbol || !idenRegex.MatchString(memberName) {
65 65                          allErrors = append(allErrors, field.Invalid(field.NewPath("spec").Child("policies"), memberName, "policy group member name is invalid"))
81 81
82 82          // Create a CEL environment with custom functions for each policy member
83 83          var opts []cel.EnvOption
84    - -       for policyName := range policyGroup.GetPolicyGroupMembers() {
   84 + +       for policyName := range policyGroup.GetPolicyGroupMembersWithContext() {
85 85                  fn := cel.Function(policyName, cel.Overload(policyName, []*cel.Type{},
```

```
                types.BoolType))
86  86              opts = append(opts, fn)
87  87          }
```

150

api/policies/v1/policygroup_validation_test.go

```
21  21                                  Name:
    "testing-cluster-policy-group",
22  22                              },
23  23                              Spec:
    ClusterAdmissionPolicyGroupSpec{
24      -
    PolicyGroupSpec: PolicyGroupSpec{
25      -
    Expression: "policy1() && policy2()",
26      -
    Message:    "This is a test policy",
27      -
    Policies: PolicyGroupMembers{
        24  +
    ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
        25  +
    GroupSpec: GroupSpec{
        26  +
    Expression: "policy1() && policy2()",
        27  +
    Message:    "This is a test policy",
        28  +                                          },
        29  +
    Policies: PolicyGroupMembersWithContext{
28      30
    "policy1": {
29      -
    Module: "ghcr.io/kubewarden/tests/user-group-
    psp:v0.4.9",
        31  +
    PolicyGroupMember: PolicyGroupMember{
        32  +
    Module: "ghcr.io/kubewarden/tests/user-group-
    psp:v0.4.9",
        33  +
    },
30      34
    },
31      35
    "policy2": {
32      -
    Module: "ghcr.io/kubewarden/tests/safe-
    labels:v1.0.0",
        36  +
    PolicyGroupMember: PolicyGroupMember{
```

```diff
+ Module: "ghcr.io/kubewarden/tests/safe-labels:v1.0.0",
+ },
  },
                                                              },
                                                      },
                                                      Name: "testing-cluster-policy-group",
                                              },
                                              Spec: ClusterAdmissionPolicyGroupSpec{
- PolicyGroupSpec: PolicyGroupSpec{
- Expression: "",
- Message:    "This is a test policy",
- Policies: PolicyGroupMembers{
+ ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
+ GroupSpec: GroupSpec{
+ Expression: "",
+ Message:    "This is a test policy",
+                                                             },
+ Policies: PolicyGroupMembersWithContext{
  "policy1": {
- Module: "ghcr.io/kubewarden/tests/user-group-psp:v0.4.9",
+ PolicyGroupMember: PolicyGroupMember{
+ Module: "ghcr.io/kubewarden/tests/user-group-psp:v0.4.9",
+ },
  },
  "policy2": {
- Module: "ghcr.io/kubewarden/tests/safe-labels:v1.0.0",
```

```diff
+ PolicyGroupMember: PolicyGroupMember{
+ Module: "ghcr.io/kubewarden/tests/safe-labels:v1.0.0",
+ },
  },
  },
  },
  Name: "testing-cluster-policy-group",
  },
  Spec: ClusterAdmissionPolicyGroupSpec{
- PolicyGroupSpec: PolicyGroupSpec{
- Expression: "123",
- Message:    "This is a test policy",
- Policies: PolicyGroupMembers{
+ ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
+ GroupSpec: GroupSpec{
+ Expression: "123",
+ Message:    "This is a test policy",
+ },
+ Policies: PolicyGroupMembersWithContext{
  "policy1": {
- Module: "ghcr.io/kubewarden/tests/user-group-psp:v0.4.9",
+ PolicyGroupMember: PolicyGroupMember{
+ Module: "ghcr.io/kubewarden/tests/user-group-psp:v0.4.9",
+ },
  },
  "policy2": {
```

```diff
- Module: "ghcr.io/kubewarden/tests/safe-labels:v1.0.0",
+ PolicyGroupMember: PolicyGroupMember{
+ Module: "ghcr.io/kubewarden/tests/safe-labels:v1.0.0",
+ },
  },
                                                             },
                                                     },
                                                     Name: "testing-cluster-policy-group",
                                             },
                                             Spec: ClusterAdmissionPolicyGroupSpec{
- PolicyGroupSpec: PolicyGroupSpec{
- Expression: "2 > 1",
- Message:    "This is a test policy",
- Policies: PolicyGroupMembers{
+ ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
+ GroupSpec: GroupSpec{
+ Expression: "2 > 1",
+ Message:    "This is a test policy",
+                                                            },
+ Policies: PolicyGroupMembersWithContext{
  "policy1": {
- Module: "ghcr.io/kubewarden/tests/user-group-psp:v0.4.9",
+ PolicyGroupMember: PolicyGroupMember{
+ Module: "ghcr.io/kubewarden/tests/user-group-psp:v0.4.9",
+ },
  },
```

```
100 122
    "policy2": {
101     -
        Module: "ghcr.io/kubewarden/tests/safe-
        labels:v1.0.0",
        123 +
        PolicyGroupMember: PolicyGroupMember{
        124 +
        Module: "ghcr.io/kubewarden/tests/safe-
        labels:v1.0.0",
        125 +
        },
102 126
    },
103 127                                                    },
104 128                                                Name:
    "testing-cluster-policy-group",
135 159                                         },
136 160                                         Spec:
    ClusterAdmissionPolicyGroupSpec{
137     -
        PolicyGroupSpec: PolicyGroupSpec{
138     -
        Expression: "policy1() && policy2()",
139     -
        Message:    "This is a test policy",
140     -
        Policies: PolicyGroupMembers{
        161 +
        ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
        162 +
        GroupSpec: GroupSpec{
        163 +
        Expression: "policy1() && policy2()",
        164 +
        Message:    "This is a test policy",
        165 +                                                },
        166 +
        Policies: PolicyGroupMembersWithContext{
141 167
        "policy1": {
142     -
        Module: "ghcr.io/kubewarden/tests/user-group-
        psp:v0.4.9",
        168 +
        PolicyGroupMember: PolicyGroupMember{
        169 +
        Module: "ghcr.io/kubewarden/tests/user-group-
        psp:v0.4.9",
        170 +
        },
```

```
143 171   },
144 172   "policy2": {
145       -   Module: "ghcr.io/kubewarden/tests/safe-labels:v1.0.0",
    173   +   PolicyGroupMember: PolicyGroupMember{
    174   +   Module: "ghcr.io/kubewarden/tests/safe-labels:v1.0.0",
    175   +   },
146 176   },
147 177                                                   },
148 178                                               },
157 187                                               Name: "testing-cluster-policy-group",
158 188                                           },
159 189                                           Spec: ClusterAdmissionPolicyGroupSpec{
160       -   PolicyGroupSpec: PolicyGroupSpec{
161       -   Policies: map[string]PolicyGroupMember{},
    190   +   ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
    191   +   Policies: map[string]PolicyGroupMemberWithContext{},
162 192                                               },
163 193                                           },
164 194                                   },
171 201                                           Name: "testing-cluster-policy-group",
172 202                                           },
173 203                                           Spec: ClusterAdmissionPolicyGroupSpec{
174       -   PolicyGroupSpec: PolicyGroupSpec{
175       -   Policies: PolicyGroupMembers{
    204   +   ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
    205   +   Policies: PolicyGroupMembersWithContext{
176 206   "": {
177       -   Module: "ghcr.io/kubewarden/tests/user-group-
```

```
                                                psp:v0.4.9",
207 +                                           PolicyGroupMember: PolicyGroupMember{
208 +                                           Module: "ghcr.io/kubewarden/tests/user-group-
                                                psp:v0.4.9",
209 +                                           },
178 210                                         },
179 211                                     },
180 212                                 },
189 221                                 Name:
                                    "testing-cluster-policy-group",
190 222                             },
191 223                             Spec:
                                    ClusterAdmissionPolicyGroupSpec{
192     -                           PolicyGroupSpec: PolicyGroupSpec{
193     -                           Policies: PolicyGroupMembers{
224 +                           ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
225 +                           Policies: PolicyGroupMembersWithContext{
194 226                             "in": {
195     -                           Module: "ghcr.io/kubewarden/tests/user-group-
                                    psp:v0.4.9",
227 +                           PolicyGroupMember: PolicyGroupMember{
228 +                           Module: "ghcr.io/kubewarden/tests/user-group-
                                    psp:v0.4.9",
229 +                           },
196 230                         },
197 231                     },
198 232                 },
207 241                 Name:
                        "testing-cluster-policy-group",
208 242             },
209 243             Spec:
                    ClusterAdmissionPolicyGroupSpec{
210     -           PolicyGroupSpec: PolicyGroupSpec{
211     -           Policies: PolicyGroupMembers{
244 +           ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
```

```diff
     245 +  Policies: PolicyGroupMembersWithContext{
212  246    "0policy1": {
213      -  Module: "ghcr.io/kubewarden/tests/user-group-psp:v0.4.9",
     247 +  PolicyGroupMember: PolicyGroupMember{
     248 +  Module: "ghcr.io/kubewarden/tests/user-group-psp:v0.4.9",
     249 +  },
214  250    },
215  251                                                  },
216  252                                          },
225  261                                          Name:
    "testing-cluster-policy-group",
226  262                                  },
227  263                                  Spec:
    ClusterAdmissionPolicyGroupSpec{
228      -  PolicyGroupSpec: PolicyGroupSpec{
229      -  Policies: PolicyGroupMembers{
     264 +  ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
     265 +  Policies: PolicyGroupMembersWithContext{
230  266    "p!ol.ic?y1": {
231      -  Module: "ghcr.io/kubewarden/tests/user-group-psp:v0.4.9",
     267 +  PolicyGroupMember: PolicyGroupMember{
     268 +  Module: "ghcr.io/kubewarden/tests/user-group-psp:v0.4.9",
     269 +  },
232  270    },
233  271                                                  },
234  272                                          },
243  281                                          Name:
    "testing-cluster-policy-group",
244  282                                  },
245  283                                  Spec:
    ClusterAdmissionPolicyGroupSpec{
```

```
246      -  PolicyGroupSpec: PolicyGroupSpec{
247      -  Policies: PolicyGroupMembers{
         284 +  ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
         285 +  Policies: PolicyGroupMembersWithContext{
248      286    "_policy1": {
249      -  Module: "ghcr.io/kubewarden/tests/user-group-psp:v0.4.9",
         287 +  PolicyGroupMember: PolicyGroupMember{
         288 +  Module: "ghcr.io/kubewarden/tests/user-group-psp:v0.4.9",
         289 +  },
250      290    },
251      291    "pol_icy2": {
252      -  Module: "ghcr.io/kubewarden/tests/safe-labels:v1.0.0",
         292 +  PolicyGroupMember: PolicyGroupMember{
         293 +  Module: "ghcr.io/kubewarden/tests/safe-labels:v1.0.0",
         294 +  },
253      295    },
254      296                                                    },
255      297                                                },
264      306                                                Name: "testing-cluster-policy-group",
265      307                                            },
266      308                                            Spec: ClusterAdmissionPolicyGroupSpec{
267      -  PolicyGroupSpec: PolicyGroupSpec{
268      -  Policies: PolicyGroupMembers{
         309 +  ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
         310 +  Policies: PolicyGroupMembersWithContext{
```

```
269 311                 "po0licy1": {
270     -               Module: "ghcr.io/kubewarden/tests/user-group-psp:v0.4.9",
    312 +               PolicyGroupMember: PolicyGroupMember{
    313 +                   Module: "ghcr.io/kubewarden/tests/user-group-psp:v0.4.9",
    314 +               },
271 315               },
272 316               "policy21": {
273     -               Module: "ghcr.io/kubewarden/tests/safe-labels:v1.0.0",
    317 +               PolicyGroupMember: PolicyGroupMember{
    318 +                   Module: "ghcr.io/kubewarden/tests/safe-labels:v1.0.0",
    319 +               },
274 320               },
275 321                                                   },
276 322                                               },
```

150 api/policies/v1/zz_generated.deepcopy.go

Load diff

21

config/crd/bases/policies.kubewarden.io_admissionpolicygroup…

```
261 261                 policies:
262 262                   additionalProperties:
263 263                     properties:
264     -                     contextAwareResources:
265     -                       description: |-
266     -                         List of Kubernetes resources the policy is allowed to access at evaluation time.
```

```
267                         - Access to these resources
    is done using the `ServiceAccount` of the
    PolicyServer
268                         - the policy is assigned to.
269                       - items:
270                         - description:
    ContextAwareResource identifies a Kubernetes
271                           - resource.
272                         - properties:
273                           - apiVersion:
274                             - description:
    apiVersion of the resource (v1 for core group,
275                           -
    groupName/groupVersions for other).
276                             - type: string
277                           - kind:
278                             - description: Singular
    PascalCase name of the resource
279                             - type: string
280                         - required:
281                         - - apiVersion
282                         - - kind
283                         - type: object
284                       - type: array
285 264                   module:
286 265                     description: |-
287 266                       Module is the location of
    the WASM module to be loaded. Can be a
```

200 docs/crds/CRD-docs-for-docs-repo.adoc

```
253 253 [cols="20a,50a,15a,15a", options="header"]
254 254 |===
255 255 | Field | Description | Default | Validation
256     - | *`PolicyGroupSpec`* __xref:{anchor_prefix}-
    github-com-kubewarden-kubewarden-controller-api-
    policies-v1-policygroupspec[$$PolicyGroupSpec$$]__
    |  |  |
    256 + | *`ClusterPolicyGroupSpec`* __xref:
    {anchor_prefix}-github-com-kubewarden-kubewarden-
    controller-api-policies-v1-
    clusterpolicygroupspec[$$ClusterPolicyGroupSpec$$]_
    _ |  |  |
257 257 | *`namespaceSelector`*
    __link:https://kubernetes.io/docs/reference/generat
    ed/kubernetes-api/v1.31/#labelselector-v1-
    meta[$$LabelSelector$$]__ | NamespaceSelector
    decides whether to run the webhook on an object
    based +
258 258 on whether the namespace for that object matches
    the selector. If the +
259 259 object itself is a namespace, the matching is
    performed on +
```

```
|===


- [id="{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-contextawareresource"]
- ==== ContextAwareResource
+ [id="{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-clusterpolicygroupspec"]
+ ==== ClusterPolicyGroupSpec
+


- ContextAwareResource identifies a Kubernetes resource.


.Appears In:
****
- - xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-clusteradmissionpolicyspec[$$ClusterAdmissionPolicySpec$$]
- - xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-policygroupmember[$$PolicyGroupMember$$]
+ - xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-clusteradmissionpolicygroupspec[$$ClusterAdmissionPolicyGroupSpec$$]
****

[cols="20a,50a,15a,15a", options="header"]
|===
| Field | Description | Default | Validation
- | *`apiVersion`* __string__ | apiVersion of the resource (v1 for core group, groupName/groupVersions for other). + |  |
- | *`kind`* __string__ | Singular PascalCase name of the resource + |  |
- |===
-
-
-
-
-
-
-
```

```
-
-
-
-
| *`GroupSpec`* __xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-groupspec[$$GroupSpec$$]__ |  |  |
| *`policies`* __xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-policygroupmemberswithcontext[$$PolicyGroupMembersWithContext$$]__ | Policies is a list of policies that are part of the group that will +
be available to be called in the evaluation expression field. +
Each policy in the group should be a Kubewarden policy. + |  | Required: {} +

|===

-[id="{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-policygroupmember"]
-==== PolicyGroupMember

+[id="{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-contextawareresource"]
+==== ContextAwareResource


+ContextAwareResource identifies a Kubernetes resource.


.Appears In:
****
- xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-policygroupmembers[$$PolicyGroupMembers$$]
+- xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-clusteradmissionpolicyspec[$$ClusterAdmissionPolicySpec$$]
+- xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-policygroupmemberwithcontext[$$PolicyGroupMemberWithContext$$]
****

[cols="20a,50a,15a,15a", options="header"]
```

```
|===
| Field | Description | Default | Validation
- | *`module`* __string__ | Module is the location of the WASM module to be loaded. Can be a +
- local file (file://), a remote file served by an HTTP server +
- (http://, https://), or an artifact served by an OCI-compatible +
- registry (registry://). +
- If prefix is missing, it will default to registry:// and use that +
- internally. + |  | Required: {} +
-
- | *`settings`* __link:https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.31/#rawextension-runtime-pkg[$$RawExtension$$]__ | Settings is a free-form object that contains the policy configuration +
- values. +
- x-kubernetes-embedded-resource: false + |  |
- | *`contextAwareResources`* __xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-contextawareresource[$$ContextAwareResource$$] array__ | List of Kubernetes resources the policy is allowed to access at evaluation time. +
- Access to these resources is done using the `ServiceAccount` of the PolicyServer +
- the policy is assigned to. + |  |
+ | *`apiVersion`* __string__ | apiVersion of the resource (v1 for core group, groupName/groupVersions for other). + |  |
+ | *`kind`* __string__ | Singular PascalCase name of the resource + |  |
|===


- [id="{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-policygroupmembers"]
- ==== PolicyGroupMembers
-
- _Underlying type:_ _xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-map-string-policygroupmember[$$map[string]PolicyGroupMember$$]_
-
-
-
-
-
```

```
- .Appears In:
- ****
- - xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-policygroupspec[$$PolicyGroupSpec$$]
- ****
-
-
-
- [id="{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-policygroupspec"]
- ==== PolicyGroupSpec
+ [id="{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-groupspec"]
+ ==== GroupSpec


  .Appears In:
  ****
- - xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-admissionpolicygroupspec[$$AdmissionPolicyGroupSpec$$]
- - xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-clusteradmissionpolicygroupspec[$$ClusterAdmissionPolicyGroupSpec$$]
+ - xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-clusterpolicygroupspec[$$ClusterPolicyGroupSpec$$]
+ - xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-policygroupspec[$$PolicyGroupSpec$$]
  ****

  [cols="20a,50a,15a,15a", options="header"]
  the policy group is rejected. The specific policy results will be +
  returned in the warning field of the response. + |
  | Required: {} +

+ |===
+
+
+
+
+
+
+
```

```
559 +
560 +
561 +
562 +
563 +
564 +
565 +
566 + [id="{anchor_prefix}-github-com-kubewarden-
    kubewarden-controller-api-policies-v1-
    policygroupmember"]
567 + ==== PolicyGroupMember
568 +
569 +
570 +
571 +
572 +
573 +
574 +
575 + .Appears In:
576 + ****
577 + - xref:{anchor_prefix}-github-com-kubewarden-
    kubewarden-controller-api-policies-v1-
    policygroupmemberwithcontext[$$PolicyGroupMemberWit
    hContext$$]
578 + - xref:{anchor_prefix}-github-com-kubewarden-
    kubewarden-controller-api-policies-v1-
    policygroupmembers[$$PolicyGroupMembers$$]
579 + ****
580 +
581 + [cols="20a,50a,15a,15a", options="header"]
582 + |===
583 + | Field | Description | Default | Validation
584 + | *`module`* __string__ | Module is the location of
    the WASM module to be loaded. Can be a +
585 + local file (file://), a remote file served by an
    HTTP server +
586 + (http://, https://), or an artifact served by an
    OCI-compatible +
587 + registry (registry://). +
588 + If prefix is missing, it will default to
    registry:// and use that +
589 + internally. + |  | Required: {} +
590 +
591 + | *`settings`*
    __link:https://kubernetes.io/docs/reference/generat
    ed/kubernetes-api/v1.31/#rawextension-runtime-
    pkg[$$RawExtension$$]__ | Settings is a free-form
    object that contains the policy configuration +
592 + values. +
593 + x-kubernetes-embedded-resource: false + |  |
594 + |===
595 +
```

```
596 +
597 + [id="{anchor_prefix}-github-com-kubewarden-
    kubewarden-controller-api-policies-v1-
    policygroupmemberwithcontext"]
598 + ==== PolicyGroupMemberWithContext
599 +
600 +
601 +
602 +
603 +
604 +
605 +
606 + .Appears In:
607 + ****
608 + - xref:{anchor_prefix}-github-com-kubewarden-
    kubewarden-controller-api-policies-v1-
    policygroupmemberswithcontext[$$PolicyGroupMembersW
    ithContext$$]
609 + ****
610 +
611 + [cols="20a,50a,15a,15a", options="header"]
612 + |===
613 + | Field | Description | Default | Validation
614 + | *`PolicyGroupMember`* __xref:{anchor_prefix}-
    github-com-kubewarden-kubewarden-controller-api-
    policies-v1-
    policygroupmember[$$PolicyGroupMember$$]__ |  |  |
615 + | *`contextAwareResources`* __xref:{anchor_prefix}-
    github-com-kubewarden-kubewarden-controller-api-
    policies-v1-
    contextawareresource[$$ContextAwareResource$$]
    array__ | List of Kubernetes resources the policy
    is allowed to access at evaluation time. +
616 + Access to these resources is done using the
    `ServiceAccount` of the PolicyServer +
617 + the policy is assigned to. + |  |
618 + |===
619 +
620 +
621 + [id="{anchor_prefix}-github-com-kubewarden-
    kubewarden-controller-api-policies-v1-
    policygroupmembers"]
622 + ==== PolicyGroupMembers
623 +
624 + _Underlying type:_ _xref:{anchor_prefix}-github-
    com-kubewarden-kubewarden-controller-api-policies-
    v1-map-string-
    policygroupmember[$$map[string]PolicyGroupMember$$]
    _
625 +
626 +
627 +
```

```
+
+
+ .Appears In:
+ ****
+ - xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-policygroupspec[$$PolicyGroupSpec$$]
+ ****
+
+
+
+ [id="{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-policygroupmemberswithcontext"]
+ ==== PolicyGroupMembersWithContext
+
+ _Underlying type:_ _xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-map-string-policygroupmemberwithcontext[$$map[string]PolicyGroupMemberWithContext$$]_
+
+
+
+
+
+ .Appears In:
+ ****
+ - xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-clusterpolicygroupspec[$$ClusterPolicyGroupSpec$$]
+ ****
+
+
+
+ [id="{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-policygroupspec"]
+ ==== PolicyGroupSpec
+
+
+
+
+
+
+
+ .Appears In:
+ ****
+ - xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-admissionpolicygroupspec[$$AdmissionPolicyGroupSpec$$]
```

```
665 + ****
666 +
667 + [cols="20a,50a,15a,15a", options="header"]
668 + |===
669 + | Field | Description | Default | Validation
670 + | *`GroupSpec`* __xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-groupspec[$$GroupSpec$$]__ | | |
587 671   | *`policies`* __xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-policygroupmembers[$$PolicyGroupMembers$$]__ | Policies is a list of policies that are part of the group that will +
588 672   be available to be called in the evaluation expression field. +
589 673   Each policy in the group should be a Kubewarden policy. + | | Required: {} +
607 691
608 692   .Appears In:
609 693   ****
610     - - xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-policygroupspec[$$PolicyGroupSpec$$]
    694 + - xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-groupspec[$$GroupSpec$$]
611 695   - xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-policyspec[$$PolicySpec$$]
612 696   ****
613 697
```

95 ■■■■□ docs/crds/CRD-docs-for-docs-repo.md

Load diff

Large diffs are not rendered by default.

26 ■■■■□

internal/controller/policyserver_controller_configmap.go

```
22 22
23 23   const dataType string = "Data" // only data type is supported
24 24
25     - type policyGroupMember struct {
    25 + type policyGroupMemberWithContext struct {
26 26           Module                  string `json:"module"`
```

```
27  27          Settings              runtime.RawExtension
    `json:"settings,omitempty"`
28  28          ContextAwareResources
    []policiesv1.ContextAwareResource
    `json:"contextAwareResources,omitempty"`
36  36          ContextAwareResources
    []policiesv1.ContextAwareResource
    `json:"contextAwareResources,omitempty"`
37  37          Settings              runtime.RawExtension
    `json:"settings,omitempty"`
38  38          // The following fields are used by policy
    groups only.
39      -       Policies   map[string]policyGroupMember
    `json:"policies,omitempty"`
40      -       Expression string
    `json:"expression,omitempty"`
41      -       Message    string
    `json:"message,omitempty"`
        39  +   Policies
    map[string]policyGroupMemberWithContext
    `json:"policies,omitempty"`
        40  +   Expression string
    `json:"expression,omitempty"`
        41  +   Message    string
    `json:"message,omitempty"`
42  42      }
43  43
44  44      // The following MarshalJSON and UnmarshalJSON
    methods are used to serialize
64  64      func (p policyServerConfigEntry) MarshalJSON()
    ([]byte, error) {
65  65          if len(p.Policies) > 0 {
66  66              return json.Marshal(struct {
67      -               NamespacedName
    types.NamespacedName
    `json:"namespacedName"`
68      -               PolicyMode     string
    `json:"policyMode"`
69      -               Policies
    map[string]policyGroupMember `json:"policies"`
70      -               Expression     string
    `json:"expression"`
71      -               Message        string
    `json:"message"`
        67  +           NamespacedName
    types.NamespacedName
    `json:"namespacedName"`
        68  +           PolicyMode     string
    `json:"policyMode"`
        69  +           Policies
    map[string]policyGroupMemberWithContext
    `json:"policies"`
```

```diff
+			Expression	string `json:"expression"`
+			Message		string `json:"message"`
		}{
			NamespacedName: p.NamespacedName,
			PolicyMode: p.PolicyMode,
	return unstructuredObj.GetResourceVersion(), nil
}

- func buildPolicyGroupMembers(policies policiesv1.PolicyGroupMembers) map[string]policyGroupMember {
-	policyGroupMembers := map[string]policyGroupMember{}
+ func buildPolicyGroupMembersWithContext(policies policiesv1.PolicyGroupMembersWithContext) map[string]policyGroupMemberWithContext {
+	policyGroupMembers := map[string]policyGroupMemberWithContext{}
	for name, policy := range policies {
-		policyGroupMembers[name] = policyGroupMember{
+		policyGroupMembers[name] = policyGroupMemberWithContext{
			Module: policy.Module,
			Settings: policy.Settings,
			ContextAwareResources: policy.ContextAwareResources,
		}

		if policyGroup, ok := admissionPolicy.(policiesv1.PolicyGroup); ok {
-			configEntry.Policies = buildPolicyGroupMembers(policyGroup.GetPolicyGroupMembers())
+			configEntry.Policies = buildPolicyGroupMembersWithContext(policyGroup.GetPolicyGroupMembersWithContext())
			configEntry.Expression = policyGroup.GetExpression()
			configEntry.Message = policyGroup.GetMessage()
		}
```

20

internal/controller/policyserver_controller_test.go

```
276 276
277 277                                clusterPolicyGroup := policiesv1.NewClusterAdmissionPolicyGroupFactory().
278 278                                WithPolicyServer(policyServerName).
279     -                              WithMembers(policiesv1.PolicyGroupMembers{
    279 +                              WithMembers(policiesv1.PolicyGroupMembersWithContext{
280 280                                "pod_privileged": {
281     -                                  Module: "registry://ghcr.io/kubewarden/tests/pod-privileged:v0.2.5",
    281 +                                  PolicyGroupMember: policiesv1.PolicyGroupMember{
    282 +                                  Module: "registry://ghcr.io/kubewarden/tests/pod-privileged:v0.2.5",
    283 +                                  },
282 284                                  ContextAwareResources: []policiesv1.ContextAwareResource{
283 285                                  {
284 286                                  APIVersion: "v1",
287 289                                  },
288 290                                  },
289 291                                "user_group_psp": {
290     -                                  Module: "registry://ghcr.io/kubewarden/tests/user-group-psp:v0.4.9",
    292 +                                  PolicyGroupMember: policiesv1.PolicyGroupMember{
    293 +                                  Module: "registry://ghcr.io/kubewarden/tests/user-group-psp:v0.4.9",
    294 +                                  },
291 295                                  ContextAwareResources: []policiesv1.ContextAwareResource{
292 296                                  {
293 297                                  APIVersion: "v1",
332 336                                  AllowedToMutate: admissionPolicyGroup.IsMutating(),
```

```
333  337                                  Settings:
                                          admissionPolicyGroup.GetSettings(),
334  338
                                          ContextAwareResources:
                                          admissionPolicyGroup.GetContextAwareResources(),
335       -                                Policies:
                                          buildPolicyGroupMembers(admissionPolicyGroup.GetPo
                                          licyGroupMembers()),
     339  +                                Policies:
                                          buildPolicyGroupMembersWithContext(admissionPolicy
                                          Group.GetPolicyGroupMembersWithContext()),
336  340                                  Expression:
                                          admissionPolicyGroup.GetExpression(),
337  341                                  Message:
                                          admissionPolicyGroup.GetMessage(),
338  342                              }
346  350                                  Settings:
                                          clusterPolicyGroup.GetSettings(),
347  351
                                          ContextAwareResources:
                                          clusterPolicyGroup.GetContextAwareResources(),
348  352                                  PolicyMode:
                                          string(clusterPolicyGroup.GetPolicyMode()),
349       -                                Policies:
                                          buildPolicyGroupMembers(clusterPolicyGroup.GetPoli
                                          cyGroupMembers()),
     353  +                                Policies:
                                          buildPolicyGroupMembersWithContext(clusterPolicyGr
                                          oup.GetPolicyGroupMembersWithContext()),
350  354                                  Expression:
                                          clusterPolicyGroup.GetExpression(),
351  355                                  Message:
                                          clusterPolicyGroup.GetMessage(),
352  356                              }
406  410
                                      }),
407  411
                                      "policies": MatchKeys(IgnoreExtras, Keys{
408  412
                                      "pod_privileged": MatchKeys(IgnoreExtras, Keys{
409       -
                                      "module":
                                      Equal(admissionPolicyGroup.GetPolicyGroupMembers()
                                      ["pod_privileged"].Module),
     413  +
                                      "module":
                                      Equal(admissionPolicyGroup.GetPolicyGroupMembersWi
                                      thContext()["pod_privileged"].Module),
410  414
                                      }),
411  415
                                  }),
```

```
412  416                "policyMode":
                        Equal(string(admissionPolicyGroup.GetPolicyMode())
                        ),
420  424            }),
421  425            "policies": MatchKeys(IgnoreExtras, Keys{
422  426                "pod_privileged": MatchAllKeys(Keys{
423       -            "module":
                        Equal(clusterPolicyGroup.GetPolicyGroupMembers()
                        ["pod_privileged"].Module),
          427  +     "module":
                        Equal(clusterPolicyGroup.GetPolicyGroupMembersWith
                        Context()["pod_privileged"].Module),
424  428                "settings": Ignore(),
425  429                "contextAwareResources":
                        And(ContainElement(MatchAllKeys(Keys{
426  430                    "apiVersion": Equal("v1"),
427  431                    "kind":       Equal("Pod"),
428  432                })), HaveLen(1)),
429  433            }),
430  434            "user_group_psp": MatchAllKeys(Keys{
431       -            "module":
                        Equal(clusterPolicyGroup.GetPolicyGroupMembers()
                        ["user_group_psp"].Module),
          435  +     "module":
                        Equal(clusterPolicyGroup.GetPolicyGroupMembersWith
                        Context()["user_group_psp"].Module),
432  436                "settings": Ignore(),
433  437                "contextAwareResources":
                        And(ContainElement(MatchAllKeys(Keys{
434  438                    "apiVersion": Equal("v1"),
```

0 comments on commit `51a88df`

Please sign in to comment.