Goals

Distribute KubeArmor policies as OCI Artifacts using registries. Store policies side by side with container images.

Proposal

Add kubearmor-client support for pushing and pulling policies to OCI registries. If needed the policy artifacts can be used in kubeArmor service itself in future.

We should own a unique artifact type for KubeArmor which must be formatted as described in OCI artifacts document.

```
application/vnd.[org|company|entity].[objectType].[optional-subType].
config.[version]+[optional-configFormat]
```

The mediaType for KuberArmor config should be:

```
application/vnd.cncf.kubearmor.config.v1+json
```

The layer mediaType for KuberArmor policy layer should be as following:

```
application/vnd.cncf.kubearmor.policy.layer.v1.yaml
application/vnd.cncf.kubearmor.policy.layer.v1.json
```

For storing more than one policy, bundle them up in `tar.gz` and use layer mediaType as:

```
application/vnd.cncf.kubearmor.policy.layer.v1.tar+gzip
```

To build this feature, we will use the ORAS client library to distribute artifacts across OCI-compliant registries.

Usage Guidelines

Run command to push a KubeArmor policy to a registry:

```
$ karmor oci push -i <img> -p <policy>
```

Run command to pull policies from a registry:

```
$ karmor oci pull -i <img> -o <dir>
```

KubeArmor policies may be pushed to an OCI-compliant registry by using the push subcommand. Use the -i flag for the image repository reference and --policy or -p to reference one or more policies which should be bundled and pushed. The -p flag supports a directory containing KubeArmor policies. The directory must only contain KubeArmor ClusterPolicy or Policy resources. Policies will be serialized and validated by the CLI first to ensure they are correct prior to pushing. This also means YAML comments will be lost. An example that pushes mypolicy.yaml along with myimage:latest image:

```
$ karmor oci push -i localhost:5000/myimage:latest -p mypolicy.yaml
```

Similar to the push subcommand, the oci command can pull the policies which were stored from a push. The -i flag is used to reference the OCI artifact representing the KubeArmor policies. The --directory or -d flag is used to set the output directory. Policies will be output as separate YAML files in the directory specified.

```
$ karmor oci pull -i localhost:5000/myimage:latest -o mydir
```

Implementation

https://github.com/akshay196/oci-kubearmor