【对外】k8s常被忽略的攻击面

一、前言

k8s有很多可以自定义扩展，然而因为开发者的安全意识缺乏导致一些问题出现，这次文章谈论的KubeApiServer扩展。

首先了解一下，KubeApiServer可以分成三种。
AggregatorServer：拦截 Aggregated APIServer 中定义的资源对象请求，并转发给相关的 Aggregated APIServer 处理。
KubeAPIServer：用于处理 k8s 的内建资源，如：Deployment，ConfigMap 等。
APIExtensionServer：负责处理用户自定义资源。

目前看到的工具大部分都是对KubeAPIServer的利用，而AggregatorServer、APIExtensionServer是没找到。自己之前苦于没有案例，所以暂时没有深入研究。

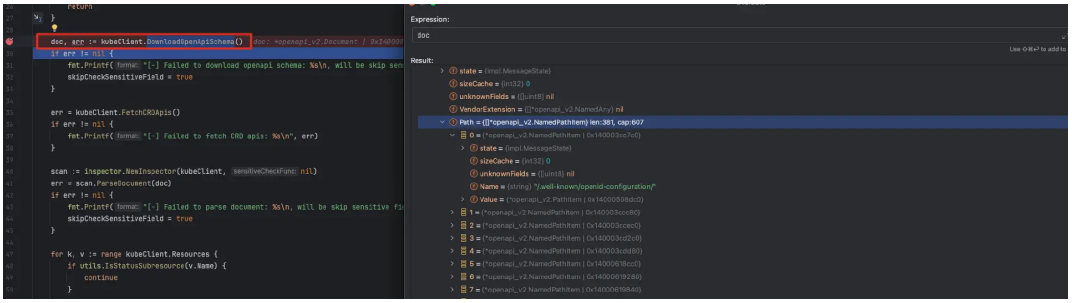
二、新的探索

最近发现有人开源了扩展KubeApi的利用工具：https://github.com/yeahx/KubeAPI-Inspector，并且提供对应的靶机环境，相当不错。

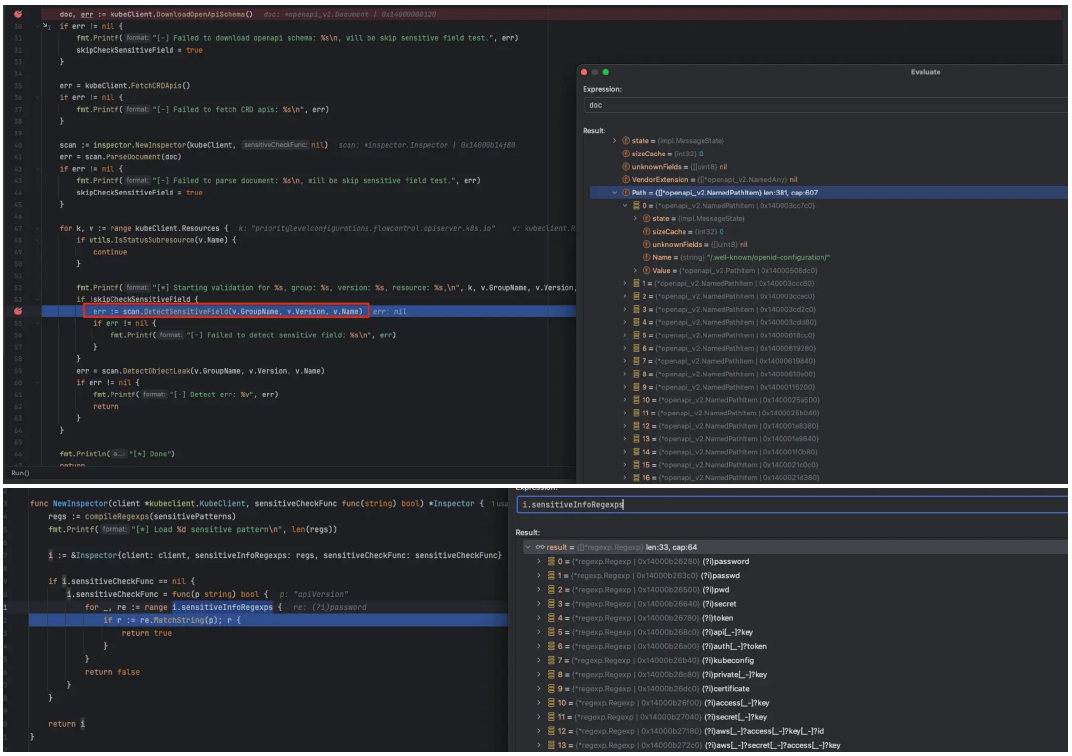
靶场演示的是注册AggregatorServer，涉及到CRD，注册 CRD 后，Kubernetes API Server 会自动提供访问该 CRD 的 RESTful 路径。例如，如果注册了 Foo CRD，Kubernetes API Server 会自动创建类似 /apis/<group>/<version>/foos 的访问路径，而无需手动注册。并且要求提供接口信息，比如字段信息。

三、查看工具利用

我们来看下工具KubeAPI-Inspector的原理。
1、下载接口schema，也就是接口信息



2、检测接口schema是否有敏感字短，比如kubeconfig



3、通过List、Watch、DeleteCollection不同的三种方法去请求。

通过不同方法想去绕过鉴权，窃取接口返回的信息。

```go
baseRes, err := i.client.List(uri)
if err != nil {
    // 403
    if !apierrors.IsForbidden(err) {
        errors = append(errors, err)
        fmt.Printf( format: "[-] verb List access apiserver failed: %v", err)
    }
}

// diff object?
// baseObj, baseLen, err := bytesToUnstructuredList(baseRes)
_, baseLen, err := utils.BytesToUnstructuredList(baseRes)
//if err != nil {
//  errors = append(errors, err)
//}

watchRes, err := i.client.Watch(uri)
if err != nil {
    if !apierrors.IsForbidden(err) {
        errors = append(errors, err)
        fmt.Printf( format: "[-] verb Watch access apiserver failed: %v", err)
    }
}

watchObj, watchLen, err := utils.WatchResToUnstructuredList(watchRes)
// diff

if watchLen > baseLen {
    utils.RemoveObjectFields(watchObj, uri)
    utils.PrintResult(utils.MakeUri(group, version, resource), verb: "Watch", watchObj)
    return nil
}

dcRes, err := i.client.DeleteCollection(uri)
if err != nil {
    if !apierrors.IsForbidden(err) {
        errors = append(errors, err)
        fmt.Printf( format: "[-] verb DeleteCollection access apiserver failed: %v", err)
    }
    // 403 skip
}

dcObj, dcLen, err := utils.BytesToUnstructuredList(dcRes)

if dcLen > baseLen {
    utils.RemoveObjectFields(dcObj, uri)
    utils.PrintResult(utils.MakeUri(group, version, resource), verb: "DeleteCollection", watchObj)
    return nil
}

// leas vres
```

这里思路和实现相对比较简单，看作者的README，以后会针对接口进行安全测试，比如owasp top 10这种漏洞。

四、总结

不仅仅kubeapiserver扩展，还有其他的扩展，我们都可以去探索下。