Findings:

1. NO\_NETWORK\_POLICY: The misconfiguration category that is related with not specifying network policies. Without specifying network policies Kubernetes installations are susceptible to unauthorized accesses.
2. NO\_ROLLING\_UPDATE: The misconfiguration category that is related with not explicitly specifying RollingUpdate in the configuration file. A lack of rolling updates makes a Kubernetes installation susceptible to supply chain related attacks.
3. INSECURE\_HTTP: : The category of using HTTP without SSL/TLS certificates to setup URLs or transmit traffic inside and outside the Kubernetes clusters.

**Info for source Code: Changed the ORG\_DIR variable to point towards the extracted folder from workshop3.zip file**

Source code for main.py:  
'''

Akond Rahman

May 31, 2021

Source Code to Run Tool on All Kubernetes Manifests

'''

import scanner

import pandas as pd

import constants

def getCountFromAnalysis(ls\_):

list2ret = []

for tup\_ in ls\_:

within\_sec\_cnt = 0

dir\_name = tup\_[0]

script\_name = tup\_[1]

within\_secret = tup\_[2] # a list of dicts: [unameDict, passwordDict, tokenDict]

within\_sec\_cnt = len(within\_secret[0]) + len( within\_secret[1] ) + len( within\_secret[2] )

'''

### format: ('data', 'password', ([], ['MTIzNAo='], [])) => (<rootKey>, <key>, <data\_list>) ... need the list of the last tuple

if isinstance( within\_secret, tuple ):

within\_sec\_cnt = len( within\_secret[-1][1] )

# print( script\_name, within\_secret, within\_sec\_cnt, type(within\_secret) )

'''

templa\_secret = tup\_[3] ### format: a list , we will not use this in dumping

taint\_secret = tup\_[4] ### format: a list

privilege\_dic = tup\_[5]

http\_dict = tup\_[6]

secuContextDic = tup\_[7]

nSpaceDict = tup\_[8]

absentResoDict = tup\_[9]

rollUpdateDic = tup\_[10]

netPolicyDict = tup\_[11]

pidfDict = tup\_[12]

ipcDict = tup\_[13]

dockersockDic = tup\_[14]

hostNetDict = tup\_[15]

cap\_sys\_dic = tup\_[16]

host\_alias\_dic = tup\_[17]

allow\_priv\_dic = tup\_[18]

unconfined\_dic = tup\_[19]

cap\_module\_dic = tup\_[20]

k8s\_flag = tup\_[21]

helm\_flag = tup\_[22]

list2ret.append( ( dir\_name, script\_name, within\_sec\_cnt, len(taint\_secret), len(privilege\_dic), len(http\_dict), len(secuContextDic), len(nSpaceDict), len(absentResoDict), len(rollUpdateDic), len(netPolicyDict), len(pidfDict), len(ipcDict), len(dockersockDic), len(hostNetDict), len(cap\_sys\_dic), len(host\_alias\_dic), len(allow\_priv\_dic), len(unconfined\_dic), len(cap\_module\_dic) , k8s\_flag, helm\_flag ) )

return list2ret

if \_\_name\_\_ == '\_\_main\_\_':

'''

DO NOT DELETE ALL IN K8S\_REPOS AS TAINT TRACKING RELIES ON BASH SCRIPTS, ONE OF THE STRENGTHS OF THE TOOL

'''

# ORG\_DIR = '/Users/arahman/K8S\_REPOS/GITHUB\_REPOS/'

# OUTPUT\_FILE\_CSV = '/Users/arahman/Documents/OneDriveWingUp/OneDrive-TennesseeTechUniversity/Research/Kubernetes/StaticTaint/data/V16\_GITHUB\_OUTPUT.csv'

# ORG\_DIR = '/Users/arahman/K8S\_REPOS/GITLAB\_REPOS/'

# OUTPUT\_FILE\_CSV = '/Users/arahman/Documents/OneDriveWingUp/OneDrive-TennesseeTechUniversity/Research/Kubernetes/StaticTaint/data/V16\_GITLAB\_OUTPUT.csv'

# ORG\_DIR = '/Users/arahman/K8S\_REPOS/BRINTO\_REPOS/'

# OUTPUT\_FILE\_CSV = '/Users/arahman/Documents/OneDriveWingUp/OneDrive-TennesseeTechUniversity/Research/Kubernetes/StaticTaint/data/V16\_BRINTO\_OUTPUT.csv'

ORG\_DIR = '/SLI-KUBE-WORK/KubeSec-master/K8S\_ARTIFACTS/'

OUTPUT\_FILE\_CSV = '/SLI-KUBE-WORK/DEMO-OUTPUT.csv'

content\_as\_ls = scanner.runScanner( ORG\_DIR )

df\_all = pd.DataFrame( getCountFromAnalysis( content\_as\_ls ) )

df\_all.to\_csv( OUTPUT\_FILE\_CSV, header= constants.CSV\_HEADER , index=False, encoding= constants.CSV\_ENCODING )