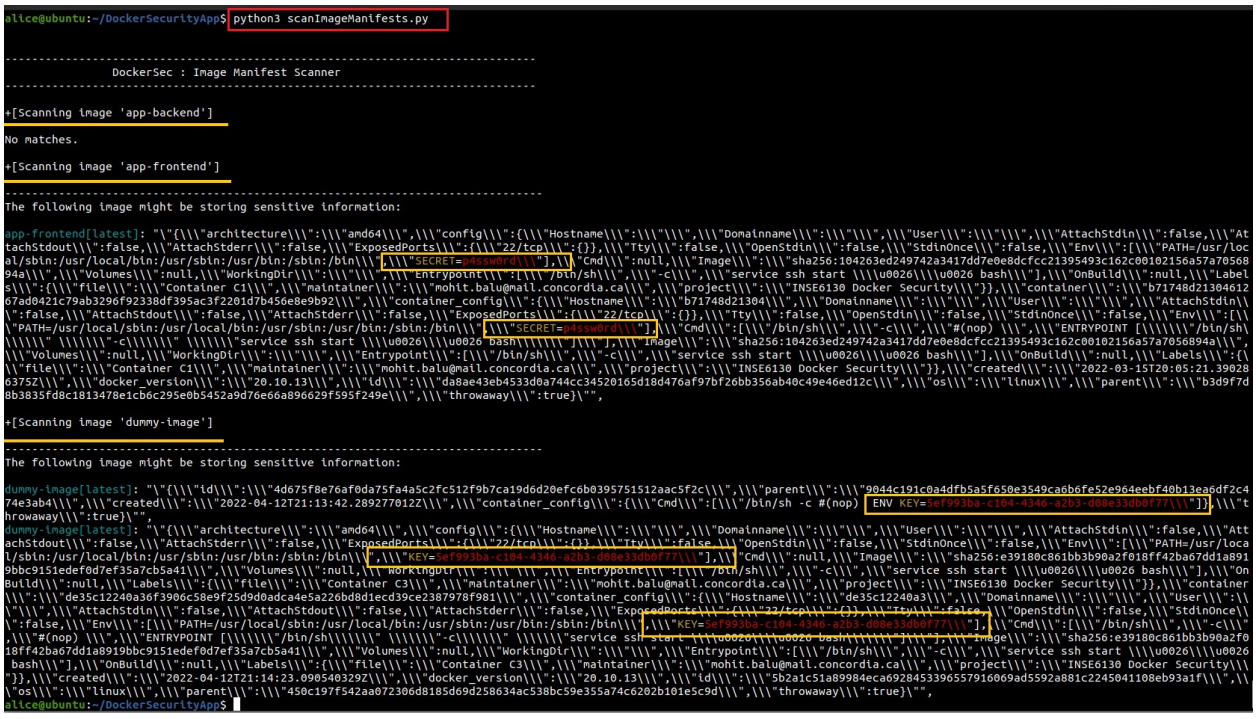
Docker Security App:

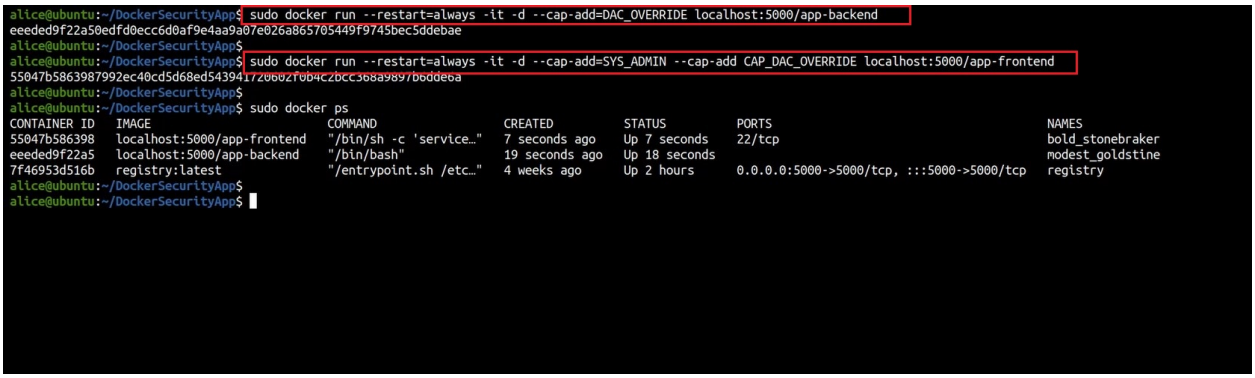
1. **Image Manifest Scanner - [Detection]**

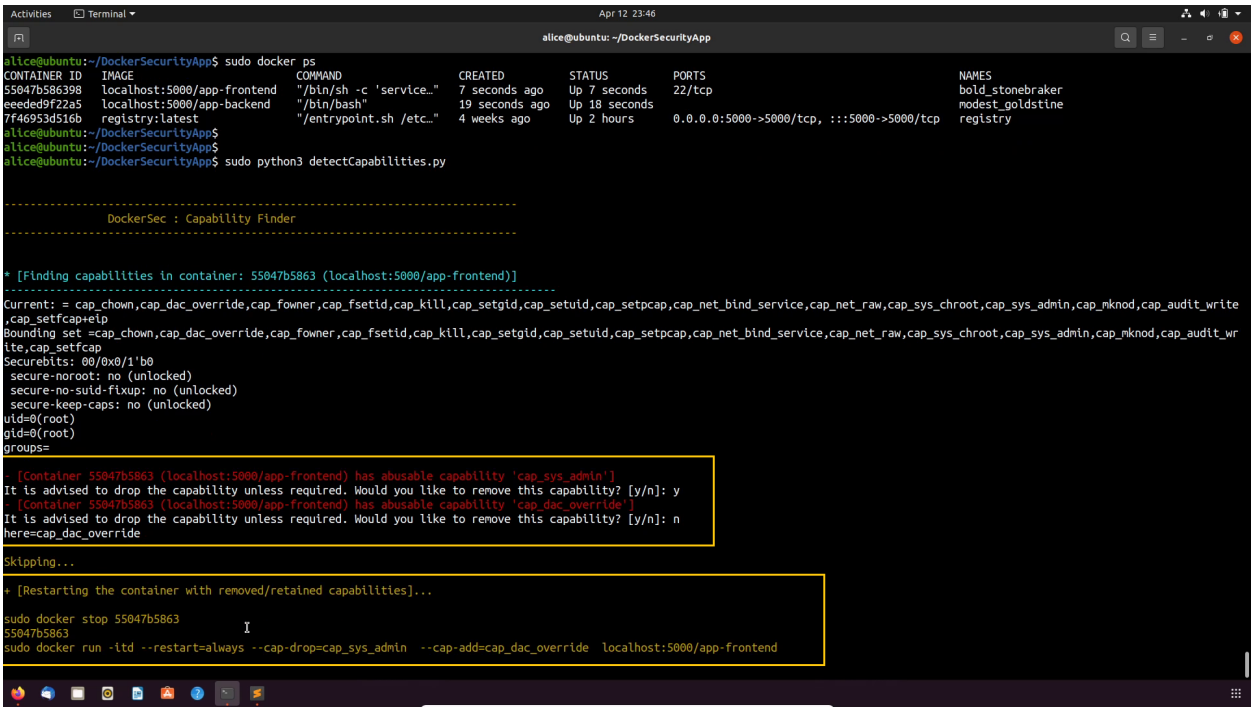
   Wrote python script scanImageManifest.py to automatically read and highlight sensitive information in image manifests from docker registry.

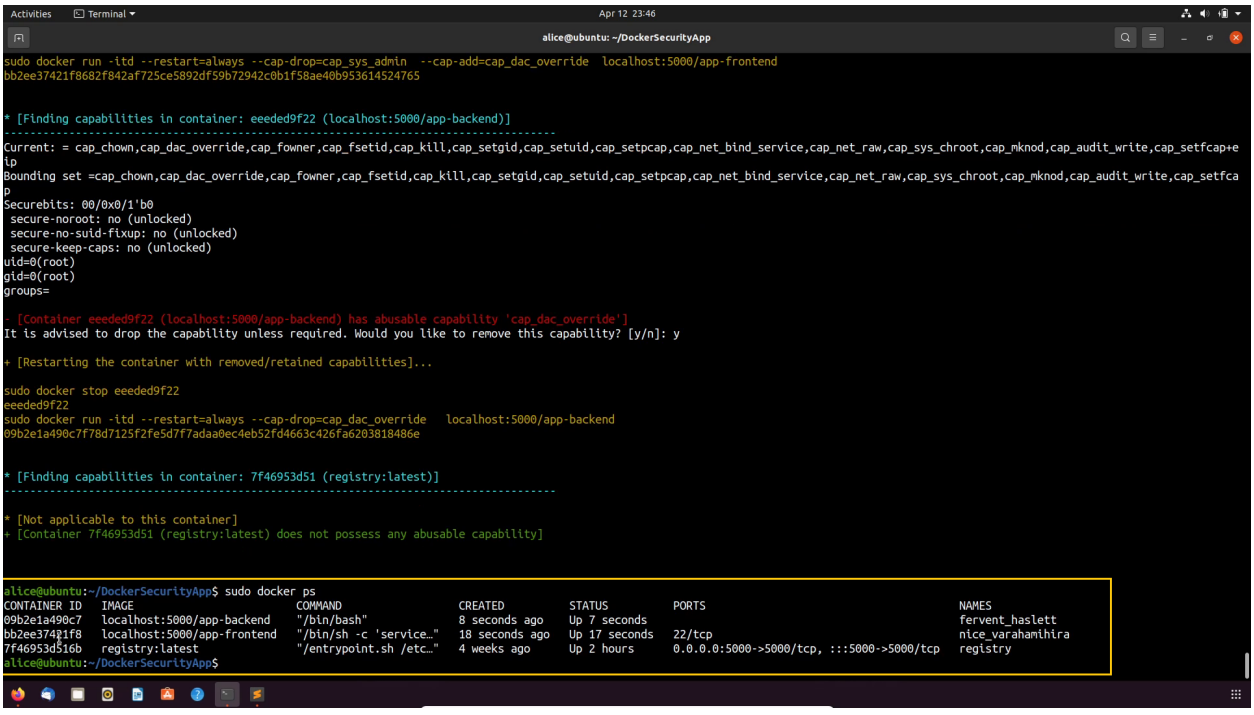


2. **Checking Capabilities - [Detection and Prevention]**

   Wrote the python script to checked if any abusable capability is present inside running containers and deleted the same.

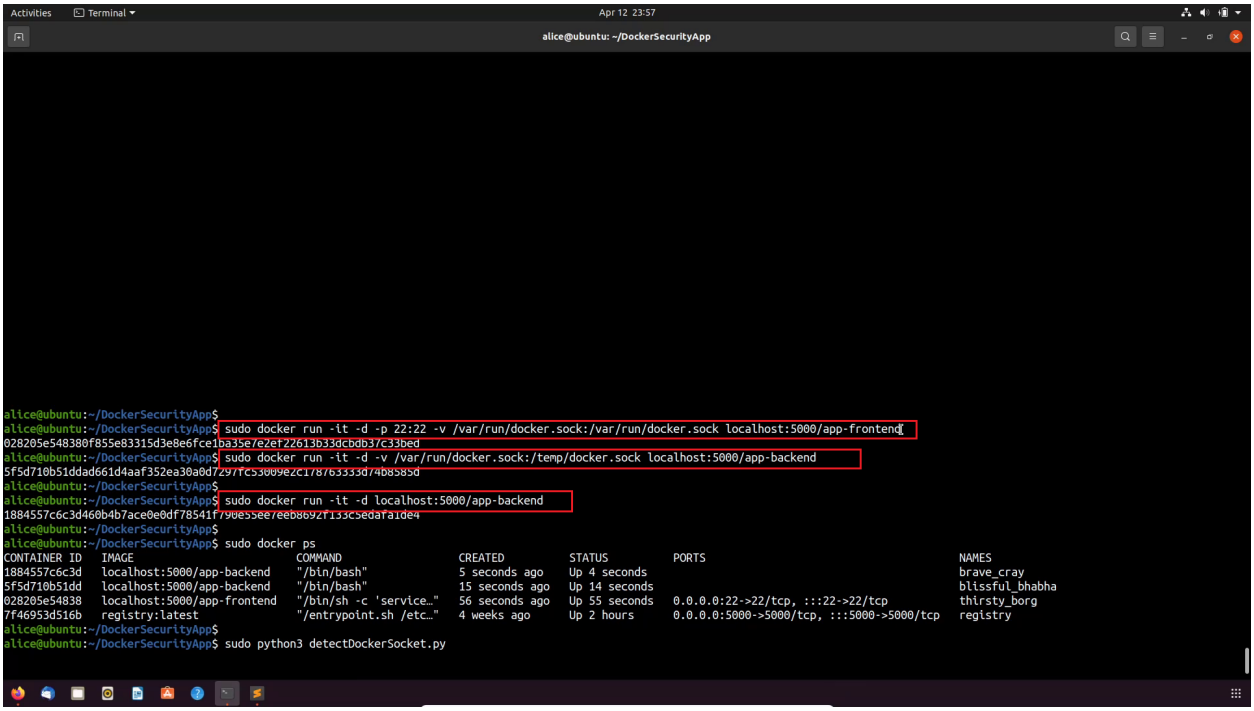


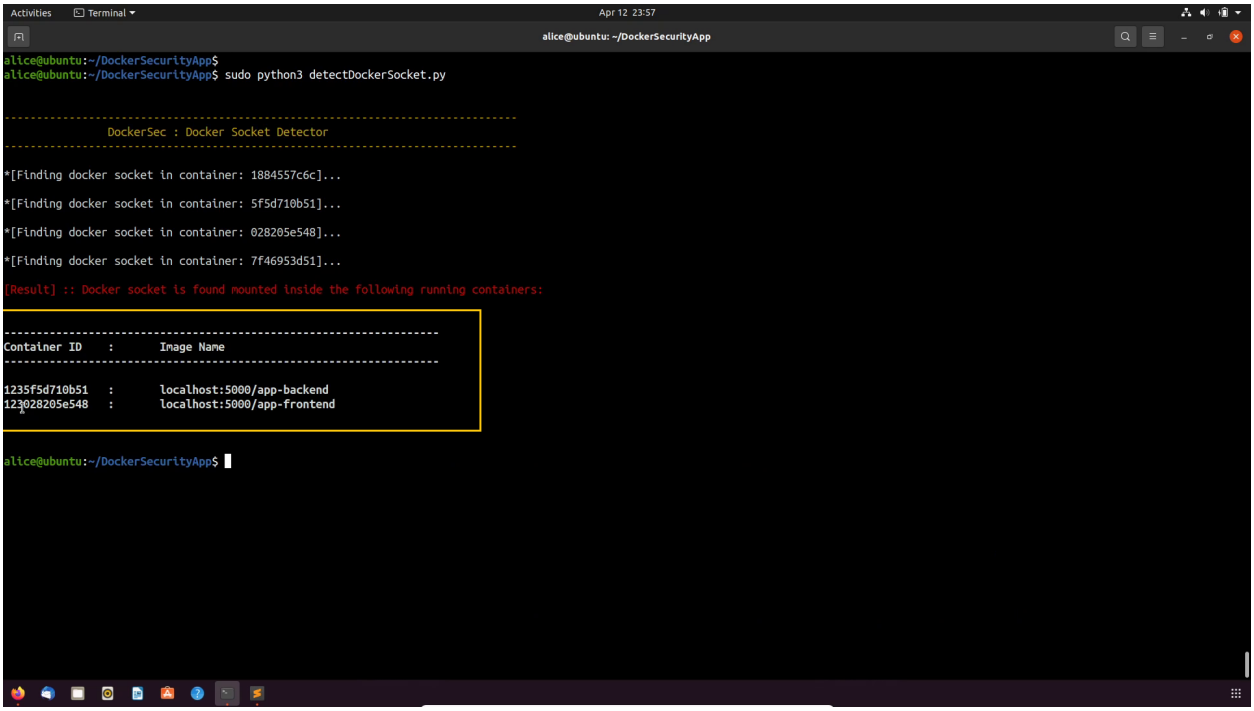

Activities
Terminal
Apr 12 23:46
alice@ubuntu: ~/DockerSecurityApp
alice@ubuntu:~/DockerSecurityApp$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
55047b586398 localhost:5000/app-frontend "/bin/sh -c 'service…" 7 seconds ago Up 7 seconds 22/tcp bold_stonebraker
eeeded9f22a5 localhost:5000/app-backend "/bin/bash" 19 seconds ago Up 18 seconds modest_goldstine
7f46953d516b registry:latest "/entrypoint.sh /etc…" 4 weeks ago Up 2 hours 0.0.0.0:5000->5000/tcp, :::5000->5000/tcp registry
alice@ubuntu:~/DockerSecurityApp$
alice@ubuntu:~/DockerSecurityApp$
alice@ubuntu:~/DockerSecurityApp$ sudo python3 detectCapabilities.py
DockerSec : Capability Finder
* [Finding capabilities in container: 55047b5863 (localhost:5000/app-frontend)]
Current: = cap_chown,cap_dac_override,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_net_bind_service,cap_net_raw,cap_sys_chroot,cap_sys_admin,cap_mknod,cap_audit_write,cap_setfcap+eip
Bounding set =cap_chown,cap_dac_override,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_net_bind_service,cap_net_raw,cap_sys_chroot,cap_sys_admin,cap_mknod,cap_audit_write,cap_setfcap
Securebits: 00/0x0/1'b0
secure-noroot: no (unlocked)
secure-no-suid-fixup: no (unlocked)
secure-keep-caps: no (unlocked)
uid=0(root)
gid=0(root)
groups=
- [Container 55047b5863 (localhost:5000/app-frontend) has abusable capability 'cap_sys_admin']
It is advised to drop the capability unless required. Would you like to remove this capability? [y/n]: y
- [Container 55047b5863 (localhost:5000/app-frontend) has abusable capability 'cap_dac_override']
It is advised to drop the capability unless required. Would you like to remove this capability? [y/n]: n
here=cap_dac_override
Skipping...
+ [Restarting the container with removed/retained capabilities]...
sudo docker stop 55047b5863
55047b5863
sudo docker run -itd --restart=always --cap-drop=cap_sys_admin --cap-add=cap_dac_override localhost:5000/app-frontend


Activities
Terminal
Apr 12 23:46
alice@ubuntu: ~/DockerSecurityApp
sudo docker run -itd --restart=always --cap-drop=cap_sys_admin --cap-add=cap_dac_override localhost:5000/app-frontend
bb2ee37421f8682f842af725ce5892df59b72942c0b1f58ae40b953614524765
* [Finding capabilities in container: eeeded9f22 (localhost:5000/app-backend)]
Current: = cap_chown,cap_dac_override,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_net_bind_service,cap_net_raw,cap_sys_chroot,cap_mknod,cap_audit_write,cap_setfcap+eip
Bounding set =cap_chown,cap_dac_override,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_net_bind_service,cap_net_raw,cap_sys_chroot,cap_mknod,cap_audit_write,cap_setfcap
Securebits: 00/0x0/1'b0
secure-noroot: no (unlocked)
secure-no-suid-fixup: no (unlocked)
secure-keep-caps: no (unlocked)
uid=0(root)
gid=0(root)
groups=
- [Container eeeded9f22 (localhost:5000/app-backend) has abusable capability 'cap_dac_override']
It is advised to drop the capability unless required. Would you like to remove this capability? [y/n]: y
+ [Restarting the container with removed/retained capabilities]...
sudo docker stop eeeded9f22
eeeded9f22
sudo docker run -itd --restart=always --cap-drop=cap_dac_override localhost:5000/app-backend
09b2e1a490c7f78d7125f2fe5d7f7adaa0ec4eb52fd4663c426fa6203818486e
* [Finding capabilities in container: 7f46953d51 (registry:latest)]
* [Not applicable to this container]
+ [Container 7f46953d51 (registry:latest) does not possess any abusable capability]
alice@ubuntu:~/DockerSecurityApp$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
09b2e1a490c7 localhost:5000/app-backend "/bin/bash" 8 seconds ago Up 7 seconds fervent_haslett
bb2ee37421f8 localhost:5000/app-frontend "/bin/sh -c 'service…" 18 seconds ago Up 17 seconds 22/tcp nice_varahamihira
7f46953d516b registry:latest "/entrypoint.sh /etc…" 4 weeks ago Up 2 hours 0.0.0.0:5000->5000/tcp, :::5000->5000/tcp registry
alice@ubuntu:~/DockerSecurityApp$

3. **Docker Socket Detector - [Detection]**

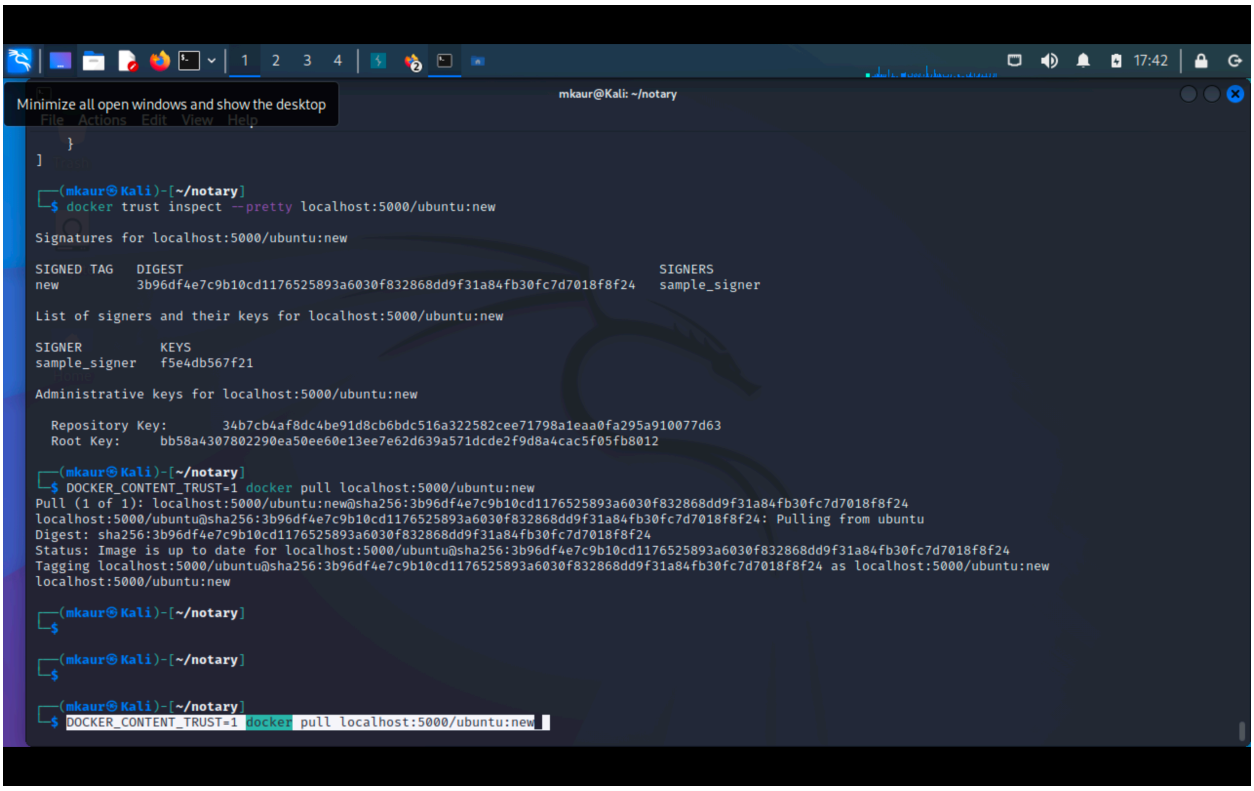
   Python script to check if 'docker.sock' file is present inside any running container.





4. **Signature Implementation - [Prevention]**

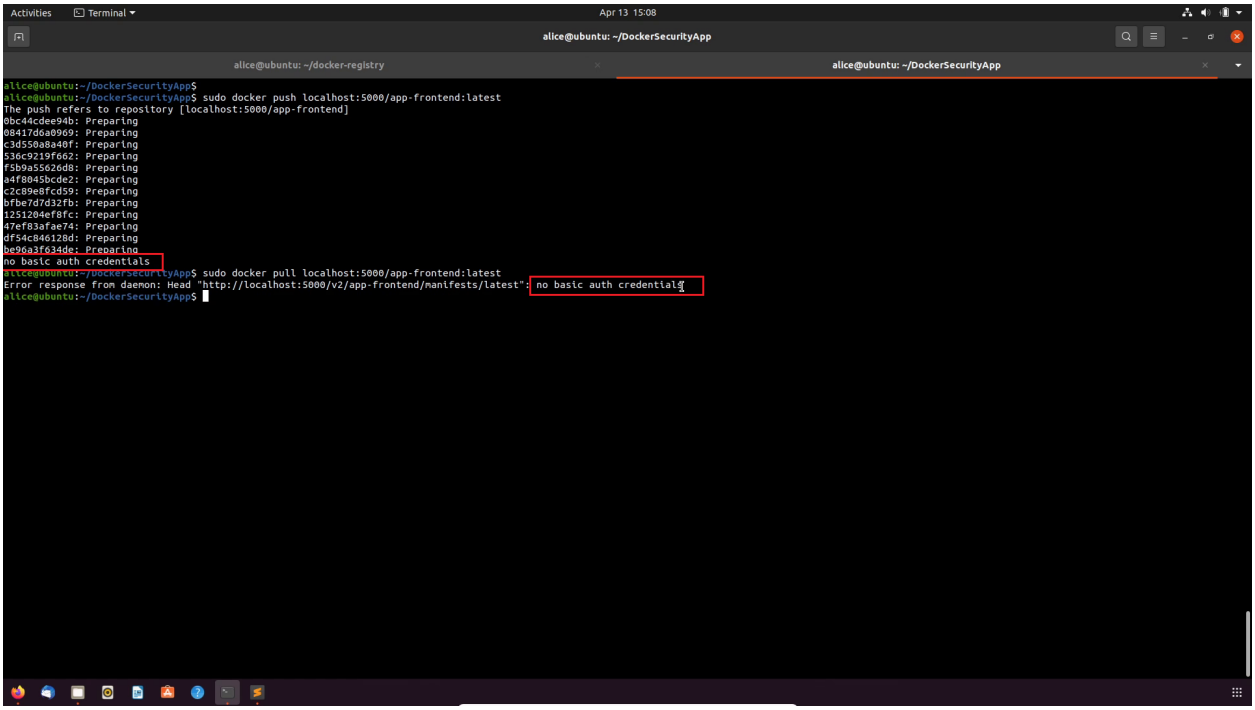
   Implemented image signature verification in docker environment and wrote python script to detect if image signature does not match.



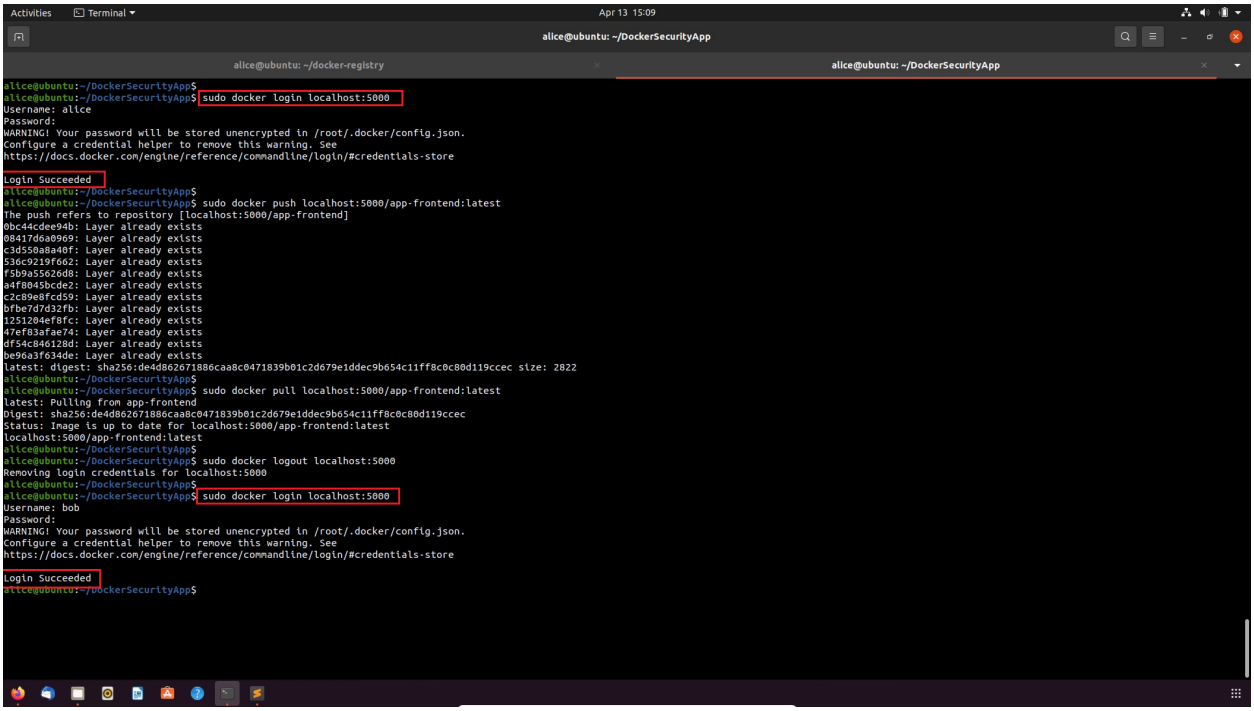
Steps:

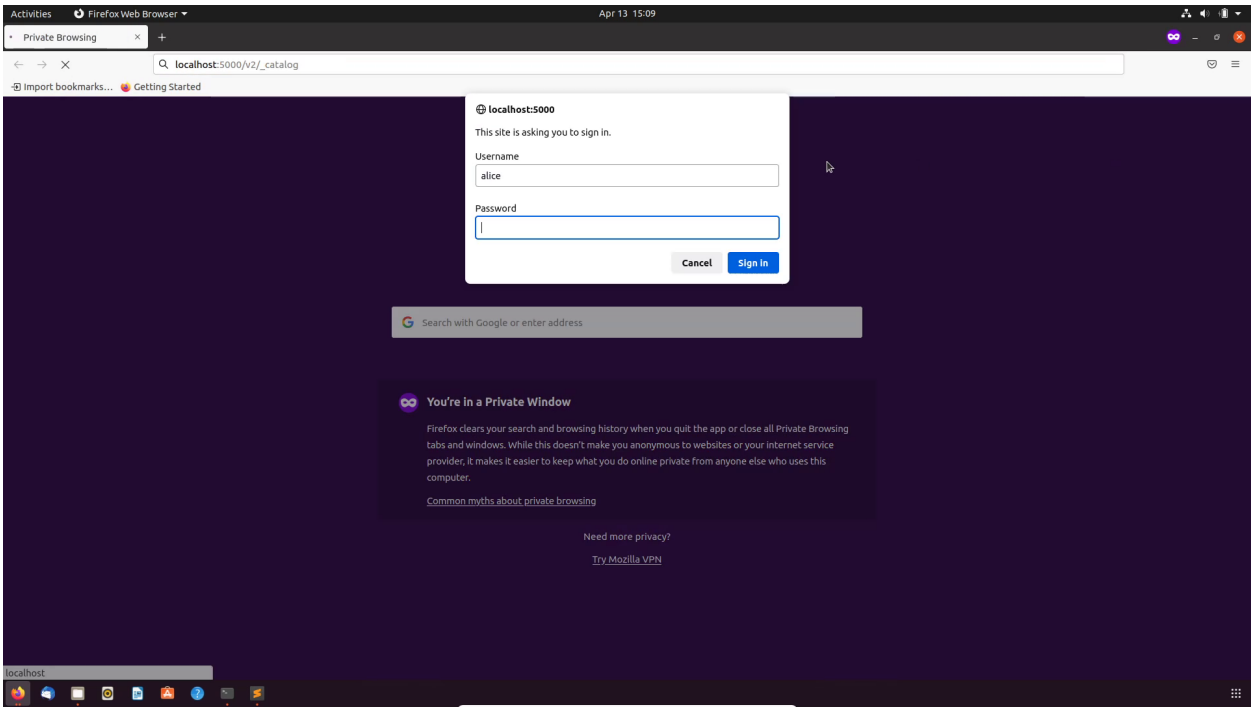
- sudo git clone https://github.com/theupdateframework/notary.git
- cd notary
- sudo docker-compose up -d
- sudo docker ps
- sudo docker pull ubuntu:new
- sudo docker run -d -p 5000:5000 —restart-always —name registry registry:latest
- export DOCKER_CONTENT_TRUST_SERVER=https://localhost:4443
- sudo docker trust key generate sample_signer
- sudo docker trust signer add --key sample_signer.pub sample_signer localhost:5000/ubuntu:new
- sudo docker trust inspect localhost:5000/ubuntu:new
- sudo docker trust sign localhost:5000/ubuntu:new
- sudo docker trust inspect --pretty localhost:5000/ubuntu:new
- DOCKER_CONTENT_TRUST=1 docker pull localhost:5000/ubuntu:new

5. **Registry Authentication - Prevention**



Implemented basic authentication on docker registry.





Steps:

- mkdir ~/docker-registry
- cd ~/docker-registry
- mkdir data
- nano docker-compose.yml

[version: '3'

services:
  registry:
    image: registry:2
    ports:
    - "5000:5000"
    environment:
      REGISTRY_AUTH: htpasswd
      REGISTRY_AUTH_HTPASSWD_REALM: Registry
      REGISTRY_AUTH_HTPASSWD_PATH: /auth/registry.password
      REGISTRY_STORAGE_FILESYSTEM_ROOTDIRECTORY: /data
    volumes:
      - ./auth:/auth
      - ./data:/data]

- docker-compose up
- sudo nano /etc/nginx/sites-available/localhost:5000

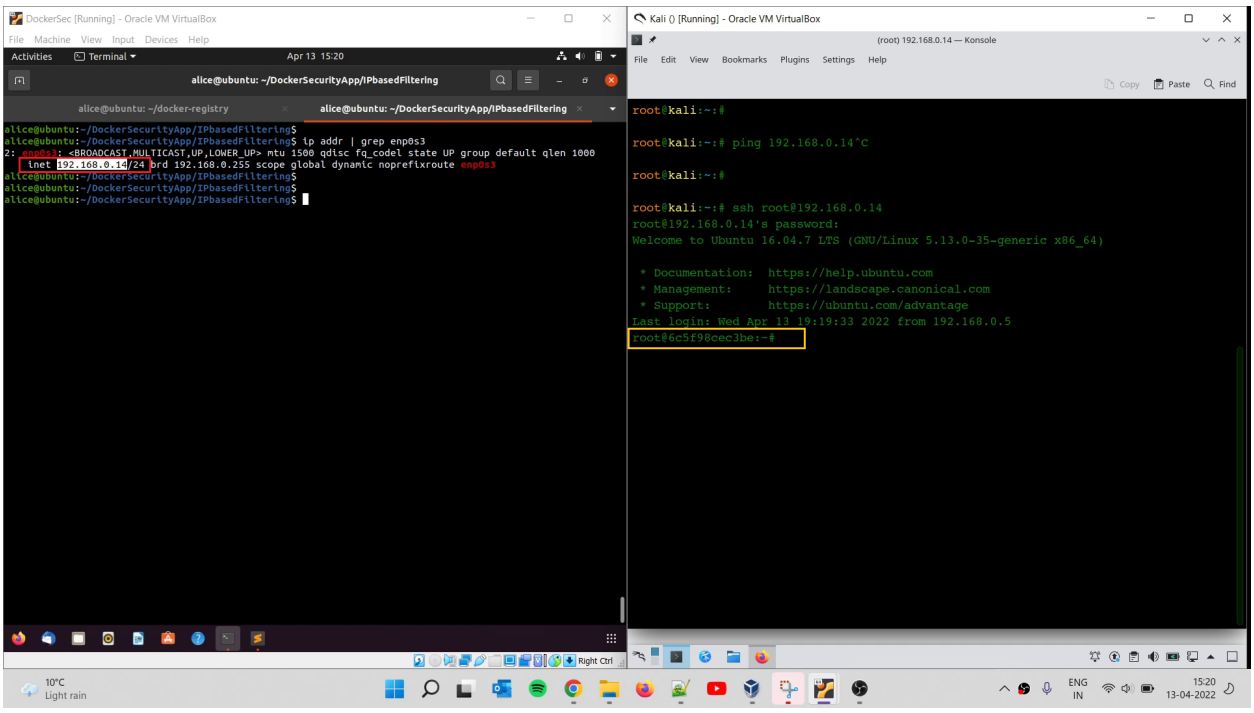
[location / {

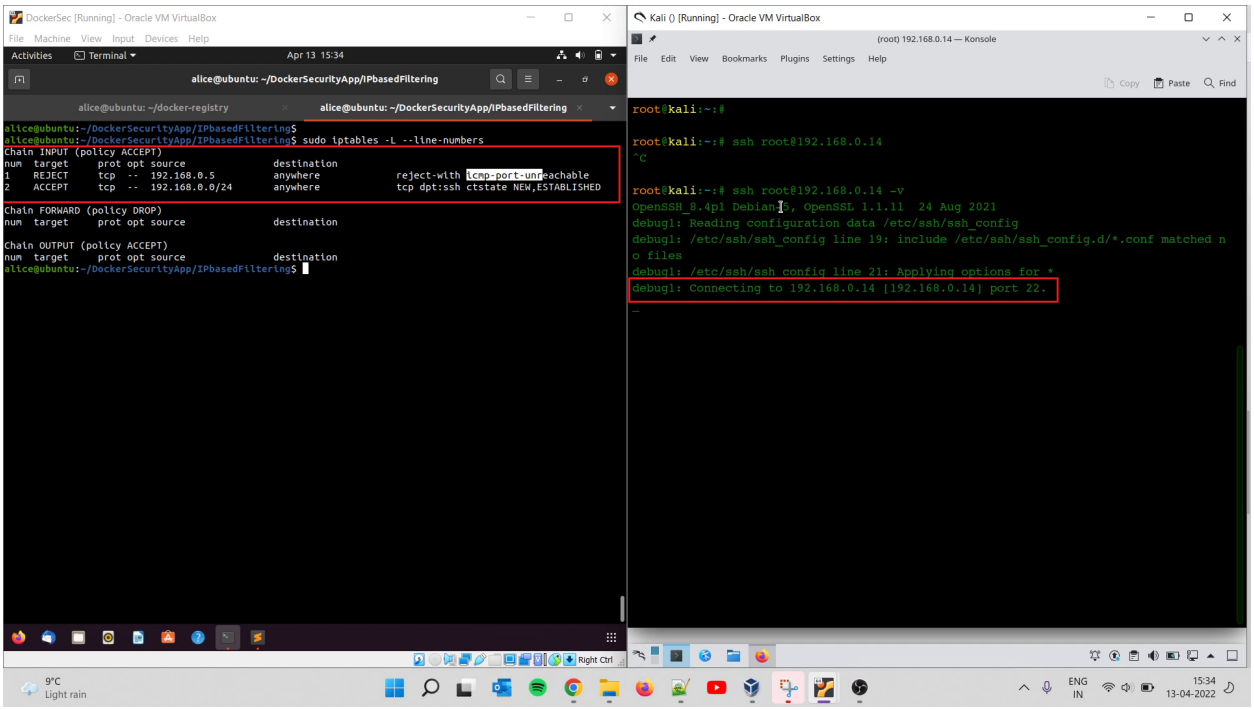
```
    # Do not allow connections from docker 1.5 and earlier
    # docker pre-1.6.0 did not properly set the user agent on ping, catch "Go *" user agents
    if ($http_user_agent ~ "^(docker\/1\.(3|4|5(?!\.[0-9]-dev))|Go ).*$" ) {
      return 404;
    }

    proxy_pass                          http://localhost:5000;
    proxy_set_header  Host              $http_host;   # required for docker client's sake
    proxy_set_header  X-Real-IP         $remote_addr; # pass on real client's IP
    proxy_set_header  X-Forwarded-For   $proxy_add_x_forwarded_for;
    proxy_set_header  X-Forwarded-Proto $scheme;
    proxy_read_timeout                  900;
}
...
```

- sudo systemctl restart nginx
- docker-compose up
- sudo apt install apache2-utils -y
- mkdir ~/docker-registry/auth
- cd ~/docker-registry/auth
- htpasswd -Bc registry.password alice
- htpasswd -B registry.password bob
- nano ~/docker-registry/docker-compose.yml
- cd ~/docker-registry
- docker-compose up

6. IP Based Filtering - [Prevention]

Implemented IP Address whitelisting on SSH login using IPtables.





Commands:
sudo iptables -L --line-numbers
sudo iptables -A INPUT -p tcp -s 192.168.0.5 -j REJECT
sudo iptables -A INPUT -p tcp -s 192.168.0.0/24 --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT