Mohammad Kaif

Remediating Publisher (Linux Machine)

Remediation

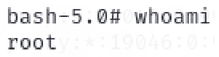
1. Publisher

1. Allowing the specific user IP to SSH

   So before we start remediating SSH, we will create a new user with sudo privileges in publisher

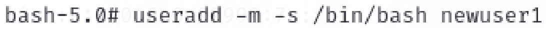
   Since we already escalated our privileges to root, from there we will be creating a new sudo user named *(newuser1)*

   - We have escalated ro root already in pen test report,



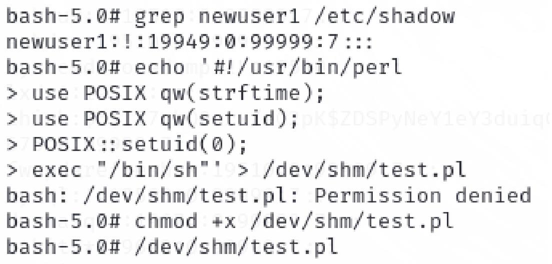
```
bash-5.0# whoami
root
```

   - Created a user name *“newuser1”*



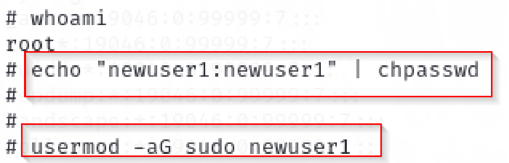
```
bash-5.0# useradd -m -s /bin/bash newuser1
```

   - We ran the apparmor bypass command so that we don’t get manipulation error



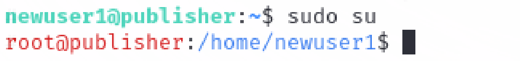
```
bash-5.0# grep newuser1 /etc/shadow
newuser1:!:19949:0:99999:7:::
bash-5.0# echo '#!/usr/bin/perl
> use POSIX qw(strftime);
> use POSIX qw(setuid);
> POSIX::setuid(0);
> exec "/bin/sh"' > /dev/shm/test.pl
bash: /dev/shm/test.pl: Permission denied
bash-5.0# chmod +x /dev/shm/test.pl
bash-5.0# /dev/shm/test.pl
```

   - We set the password to “newuser1”, and added him to sudo lists.



```
# whoami
root
# echo "newuser1:newuser1" | chpasswd
#
#
# usermod -aG sudo newuser1
```

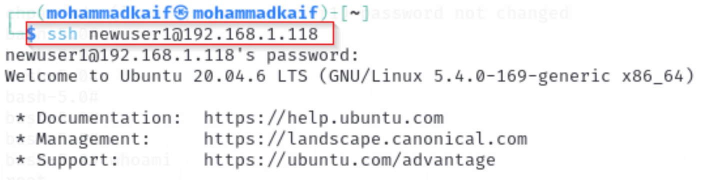
   - And we were able to SSH through that *newuser1*, and access root privileges by sudo su



```
newuser1@publisher:~$ sudo su
root@publisher:/home/newuser1$
```

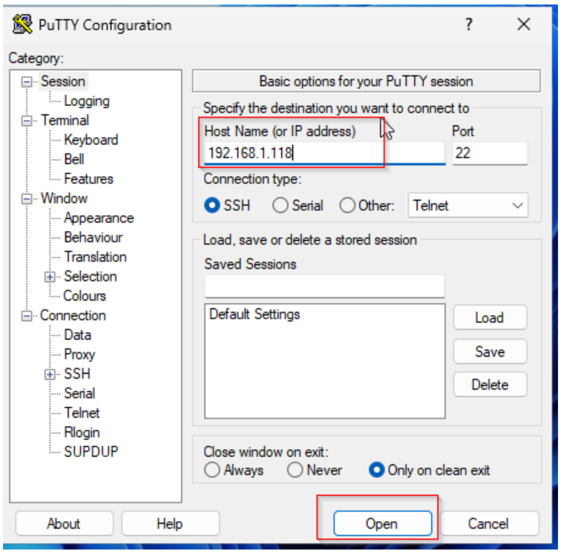
   - Now we are going to give access to specific IP address by editing the SSH configuration file.
   - Open the SSH config file using sudo nano *‘/etc/ssh/sshd_config’*
   - Now we wrote ‘*AllowUsers newuser1@192.168.1.106*’. By using this, we allowed newuser1 to access only from the Kali machine whose IP is 192.168.1.106.

   - Since we specified the kali machine IP we could SSH through it.



```
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ ssh newuser1@192.168.1.118
newuser1@192.168.1.118's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-169-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
```

   - However, we tried to get SSH from a Windows machine using Putty, but it said *access denied*.



Specify the destination you want to connect to

Host Name (or IP address): 192.168.1.118

Port: 22

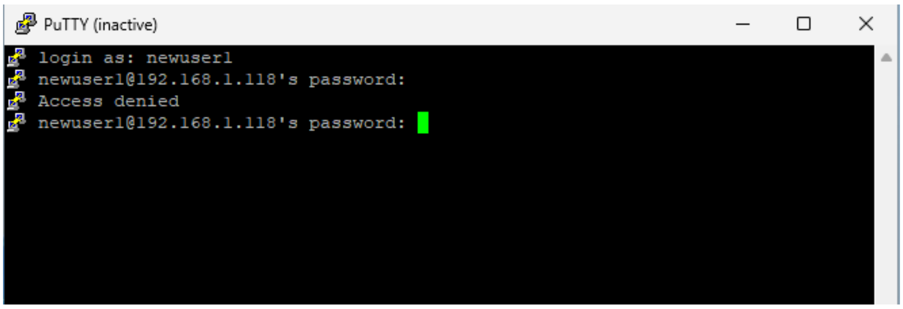
Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

Saved Sessions

Default Settings

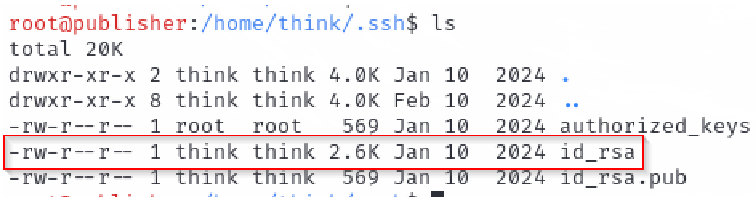
Close window on exit: ○ Always ○ Never ● Only on clean exit



```
login as: newuser1
newuser1@192.168.1.118's password:
Access denied
newuser1@192.168.1.118's password:
```

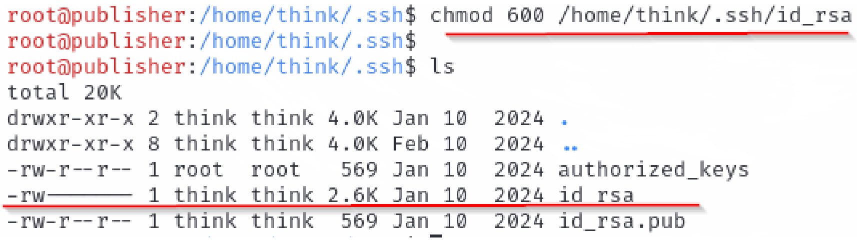
2. <u>Restric permissions and access to SSH keys</u>

- During our exploitations we were able to get through a shell as *www.data* user and can view the id_rsa of think. This is because wrong permissions are set on the id_rsa of the user think as we can see from the screenshot. The *-rw-r—r—* shows that the root and other users have read and write permission to id_rsa of *think*.



```
root@publisher:/home/think/.ssh$ ls
total 20K
drwxr-xr-x 2 think think 4.0K Jan 10  2024 .
drwxr-xr-x 8 think think 4.0K Feb 10  2024 ..
-rw-r--r-- 1 root  root   569 Jan 10  2024 authorized_keys
-rw-r--r-- 1 think think 2.6K Jan 10  2024 id_rsa
-rw-r--r-- 1 think think  569 Jan 10  2024 id_rsa.pub
```

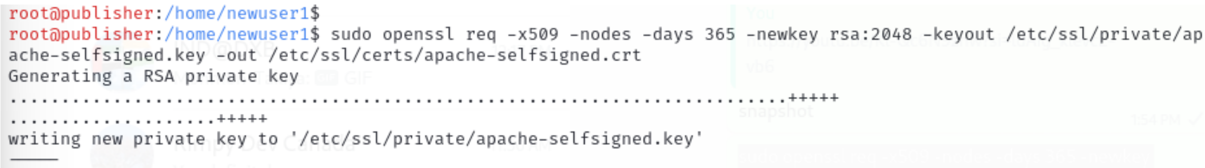
- So, we changed the permission on id_rsa file to , this ensures that only the *think* user can access his/her private key



```
root@publisher:/home/think/.ssh$ chmod 600 /home/think/.ssh/id_rsa
root@publisher:/home/think/.ssh$
root@publisher:/home/think/.ssh$ ls
total 20K
drwxr-xr-x 2 think think 4.0K Jan 10  2024 .
drwxr-xr-x 8 think think 4.0K Feb 10  2024 ..
-rw-r--r-- 1 root  root   569 Jan 10  2024 authorized_keys
-rw------- 1 think think 2.6K Jan 10  2024 id_rsa
-rw-r--r-- 1 think think  569 Jan 10  2024 id_rsa.pub
```

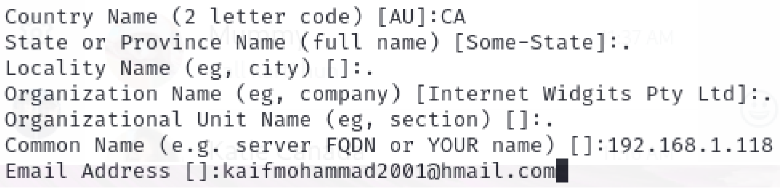
3. <u>Remediating http to https</u>

- Generating the SSL self signed certificate on publisher, as e have ssh on it. In this stage, you provide the data required to create a self-signed SSL certificate. In order to ensure that clients connecting to the server can authenticate it, the information entered helps uniquely identify the server and its owner. In order to establish secure communication via HTTPS, this information is essential



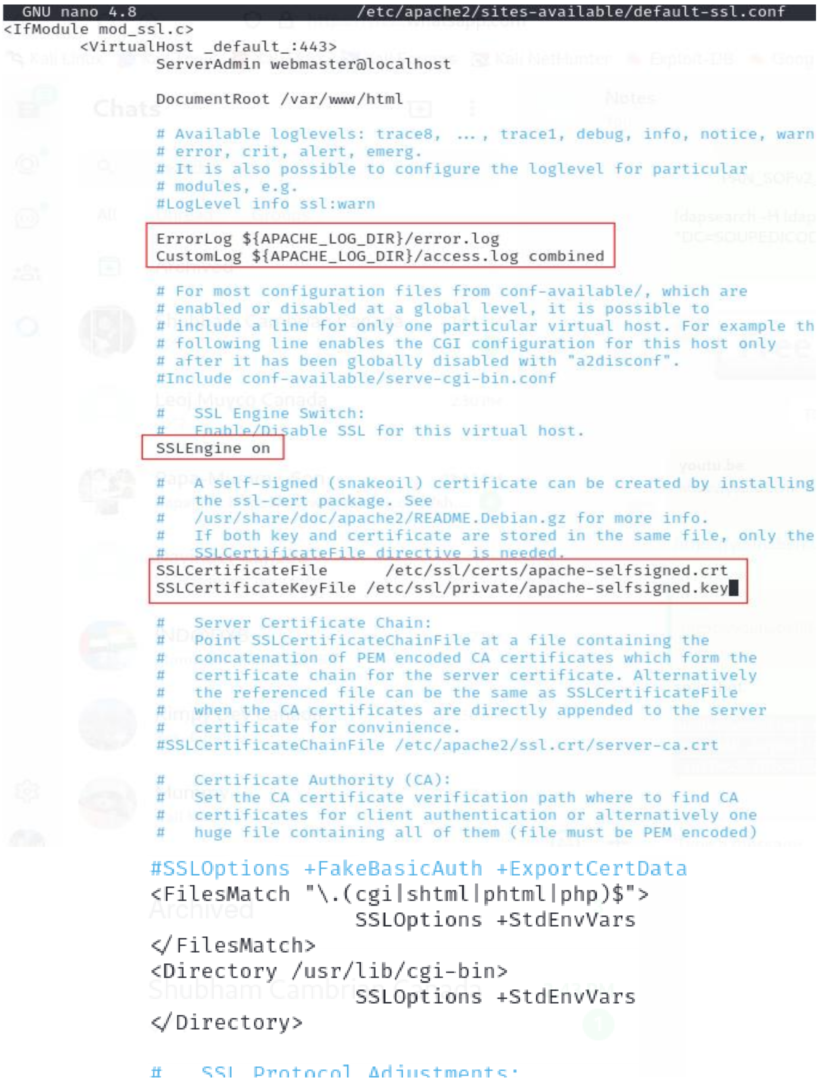
```
root@publisher:/home/newuser1$
root@publisher:/home/newuser1$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/ap
ache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
Generating a RSA private key
.........................................................................+++++
....................+++++
writing new private key to '/etc/ssl/private/apache-selfsigned.key'
-----
```

- In this stage, you provide the data required to create a self-signed SSL certificate. In order to ensure that clients connecting to the server can authenticate it, the information entered helps uniquely identify the server and its owner. In order to establish secure communication via HTTPS, this information is essential.



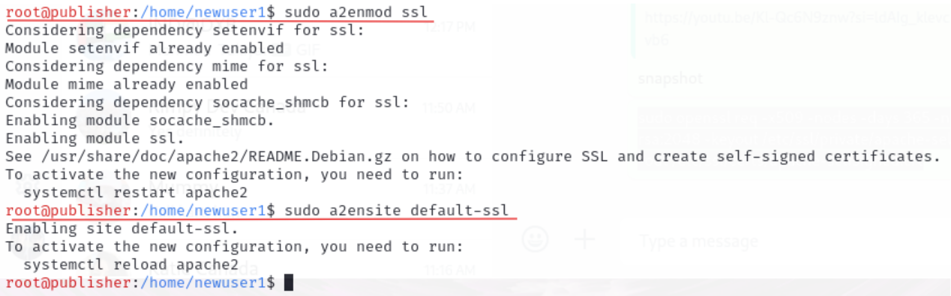
```
Country Name (2 letter code) [AU]:CA
State or Province Name (full name) [Some-State]:.
Locality Name (eg, city) []:.
Organization Name (eg, company) [Internet Widgits Pty Ltd]:.
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:192.168.1.118
Email Address []:kaifmohammad2001@hmail.com
```

- Configure the default-ssl.config file
    - Virtual host that listens on port 443 is defined here.
    - indicates the file that contains the access and error messages linked to this
    - virtual host.
    - For this virtual host, turns on SSL.
    - Both the private key and SSL certificate are mentioned.
    - Security is improved through the configuration of the protocol and cipher



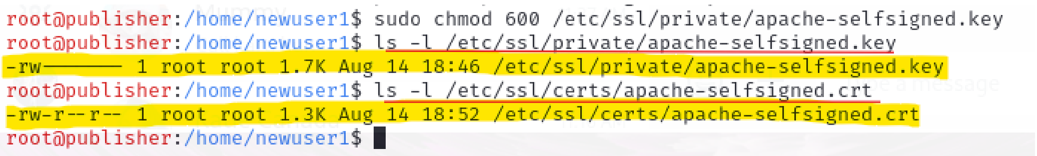
```
  GNU nano 4.8                /etc/apache2/sites-available/default-ssl.conf
<IfModule mod_ssl.c>
        <VirtualHost _default_:443>
                ServerAdmin webmaster@localhost

                DocumentRoot /var/www/html

                # Available loglevels: trace8, ..., trace1, debug, info, notice, warn
                # error, crit, alert, emerg.
                # It is also possible to configure the loglevel for particular
                # modules, e.g.
                #LogLevel info ssl:warn

                ErrorLog ${APACHE_LOG_DIR}/error.log
                CustomLog ${APACHE_LOG_DIR}/access.log combined

                # For most configuration files from conf-available/, which are
                # enabled or disabled at a global level, it is possible to
                # include a line for only one particular virtual host. For example th
                # following line enables the CGI configuration for this host only
                # after it has been globally disabled with "a2disconf".
                #Include conf-available/serve-cgi-bin.conf

                #   SSL Engine Switch:
                #   Enable/Disable SSL for this virtual host.
                SSLEngine on

                #   A self-signed (snakeoil) certificate can be created by installing
                #   the ssl-cert package. See
                #   /usr/share/doc/apache2/README.Debian.gz for more info.
                #   If both key and certificate are stored in the same file, only the
                #   SSLCertificateFile directive is needed.
                SSLCertificateFile      /etc/ssl/certs/apache-selfsigned.crt
                SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

                #   Server Certificate Chain:
                #   Point SSLCertificateChainFile at a file containing the
                #   concatenation of PEM encoded CA certificates which form the
                #   certificate chain for the server certificate. Alternatively
                #   the referenced file can be the same as SSLCertificateFile
                #   when the CA certificates are directly appended to the server
                #   certificate for convinience.
                #SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

                #   Certificate Authority (CA):
                #   Set the CA certificate verification path where to find CA
                #   certificates for client authentication or alternatively one
                #   huge file containing all of them (file must be PEM encoded)
                #SSLOptions +FakeBasicAuth +ExportCertData
                <FilesMatch "\.(cgi|shtml|phtml|php)$">
                                SSLOptions +StdEnvVars
                </FilesMatch>
                <Directory /usr/lib/cgi-bin>
                                SSLOptions +StdEnvVars
                </Directory>

                #   SSL Protocol Adjustments:
```

- Now enabling the SSL module as well as the SSL site module In this step, Apache's default-ssl site configuration and the SSL module are enabled. By turning them on, you can be sure that Apache can manage SSL/TLS traffic with the configuration options found in the default-ssl.conf file. Once the Apache service has been restarted or reloaded, the modifications will take effect.



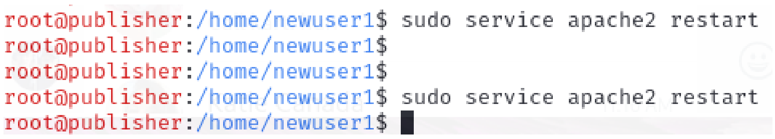
```
root@publisher:/home/newuser1$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@publisher:/home/newuser1$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@publisher:/home/newuser1$ 
```

- We have created the key and the self-signed SSL certificate, as shown in the screenshots, and they are both in the appropriate directories with the appropriate permissions.



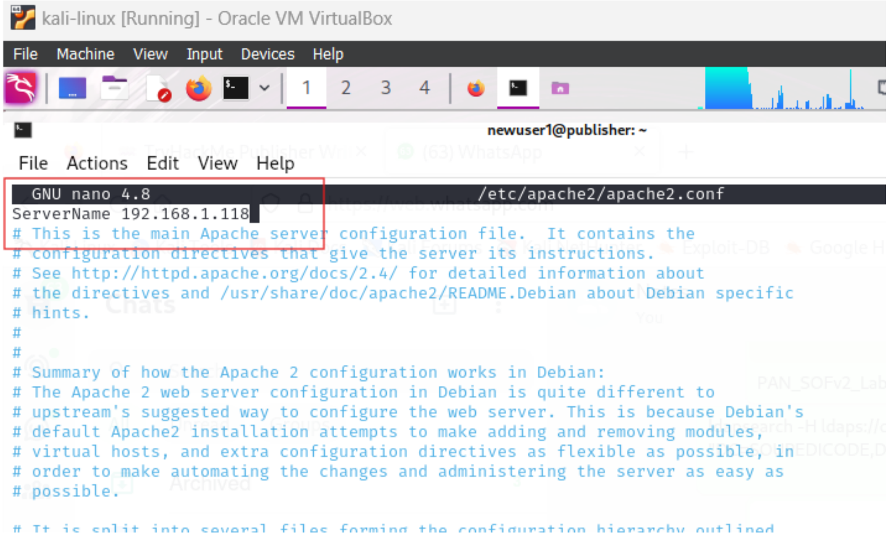
```
root@publisher:/home/newuser1$ sudo chmod 600 /etc/ssl/private/apache-selfsigned.key
root@publisher:/home/newuser1$ ls -l /etc/ssl/private/apache-selfsigned.key
-rw------- 1 root root 1.7K Aug 14 18:46 /etc/ssl/private/apache-selfsigned.key
root@publisher:/home/newuser1$ ls -l /etc/ssl/certs/apache-selfsigned.crt
-rw-r--r-- 1 root root 1.3K Aug 14 18:52 /etc/ssl/certs/apache-selfsigned.crt
root@publisher:/home/newuser1$
```

- Now restart the apache service



```
root@publisher:/home/newuser1$ sudo service apache2 restart
root@publisher:/home/newuser1$
root@publisher:/home/newuser1$
root@publisher:/home/newuser1$ sudo service apache2 restart
root@publisher:/home/newuser1$
```

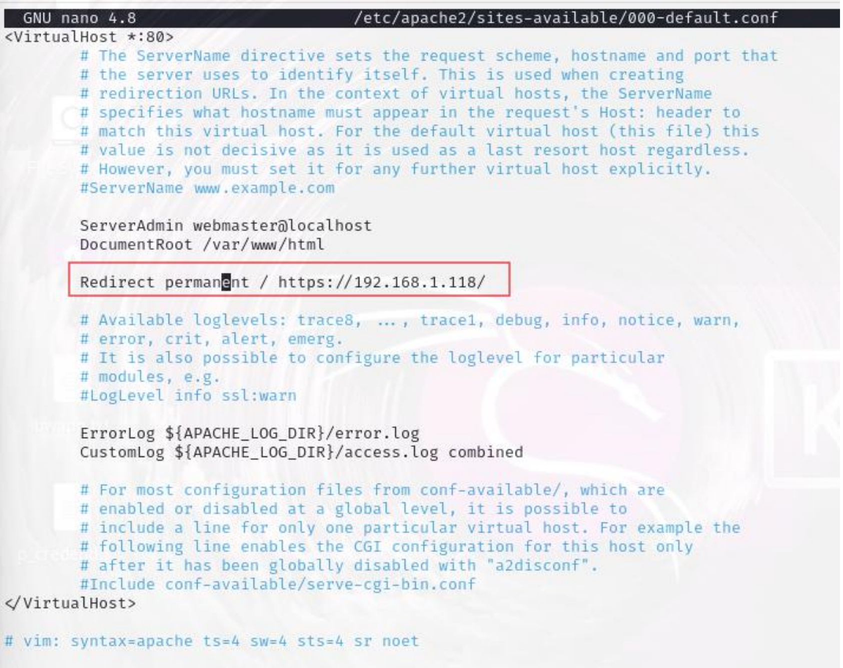
- Now open the main Apache config file and added the IP of Publisher in the ‘server name’ directive. We make sure that Apache recognizes Publisher's IP address as the server's identity by setting the ServerName to that address. This avoids possible warnings and facilitates accurate server identification.



kali-linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
newuser1@publisher: ~
File Actions Edit View Help

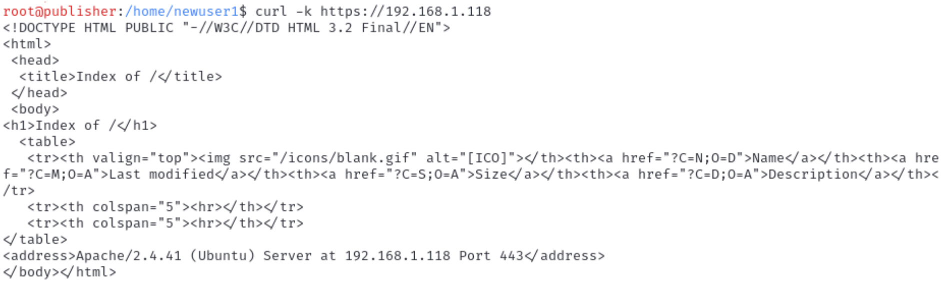
```
  GNU nano 4.8                                /etc/apache2/apache2.conf
ServerName 192.168.1.118
# This is the main Apache server configuration file.  It contains the
# configuration directives that give the server its instructions.
# See http://httpd.apache.org/docs/2.4/ for detailed information about
# the directives and /usr/share/doc/apache2/README.Debian about Debian specific
# hints.
#
#
# Summary of how the Apache 2 configuration works in Debian:
# The Apache 2 web server configuration in Debian is quite different to
# upstream's suggested way to configure the web server. This is because Debian's
# default Apache2 installation attempts to make adding and removing modules,
# virtual hosts, and extra configuration directives as flexible as possible, in
# order to make automating the changes and administering the server as easy as
# possible.

# It is split into several files forming the configuration hierarchy outlined
```

- Directing all traffic to https. We have configured a redirection from HTTP to HTTPS in this configuration. This is accomplished by creating a virtual host that is listening on port 80 and redirecting all traffic to the HTTPS URL with the 'Redirect directive'. Making sure that every connection to your server is encrypted and secure requires doing this step



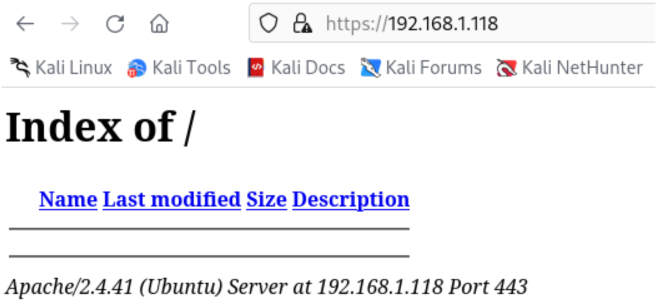
```
  GNU nano 4.8                          /etc/apache2/sites-available/000-default.conf
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        Redirect permanent / https://192.168.1.118/

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

- The fact that the connection is over HTTPS (port 443) shows that SSL/TLS is configured correctly; but, since we are using curl with the -k parameter, we are not taking SSL certificate verification problems into account.



```
root@publisher:/home/newuser1$ curl -k https://192.168.1.118
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
 <head>
  <title>Index of /</title>
 </head>
 <body>
<h1>Index of /</h1>
  <table>
   <tr><th valign="top"><img src="/icons/blank.gif" alt="[ICO]"></th><th><a href="?C=N;O=D">Name</a></th><th><a hre
f="?C=M;O=A">Last modified</a></th><th><a href="?C=S;O=A">Size</a></th><th><a href="?C=D;O=A">Description</a></th><
/tr>
   <tr><th colspan="5"><hr></th></tr>
   <tr><th colspan="5"><hr></th></tr>
</table>
<address>Apache/2.4.41 (Ubuntu) Server at 192.168.1.118 Port 443</address>
</body></html>
```

- The page also indicates that it is being served on Port 443, the default HTTPS traffic port. However, a little alert sign suggests that there is a potential issue with the SSL/TLS certificate.
- We rechecked and redo the steps but couldn't fix the ssl error.
- Also tried regenerating the SSL certificate.

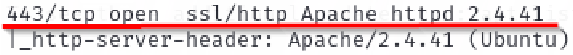


Index of /

Name Last modified Size Description

Apache/2.4.41 (Ubuntu) Server at 192.168.1.118 Port 443

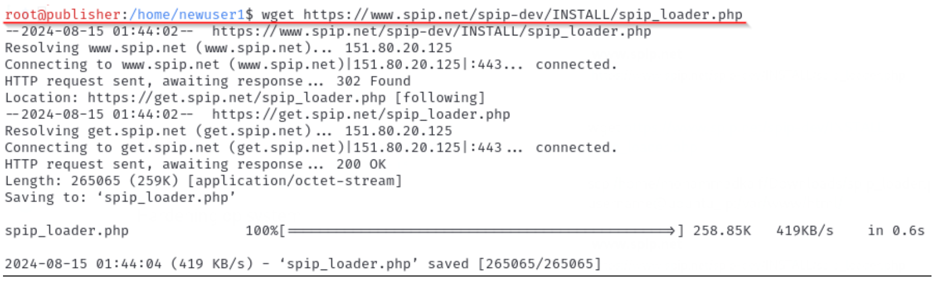
- We can confirm that its listening on port 443 instead of 80



```
443/tcp open  ssl/http Apache httpd 2.4.41
|_http-server-header: Apache/2.4.41 (Ubuntu)
```

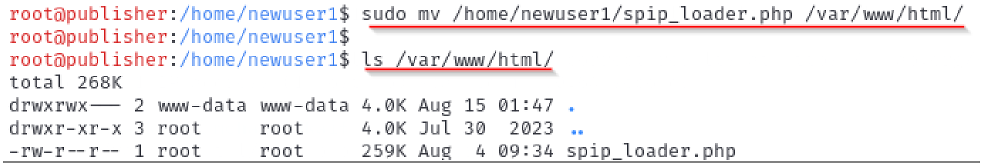
4. <u>Remediating SPIP</u>

- Download the *spip_loader.php*



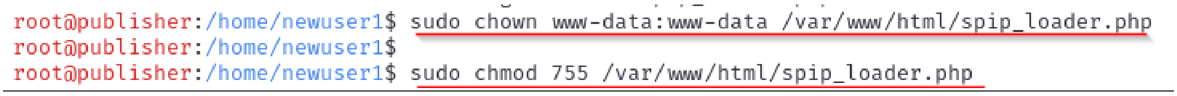
```
root@publisher:/home/newuser1$ wget https://www.spip.net/spip-dev/INSTALL/spip_loader.php
--2024-08-15 01:44:02--  https://www.spip.net/spip-dev/INSTALL/spip_loader.php
Resolving www.spip.net (www.spip.net)... 151.80.20.125
Connecting to www.spip.net (www.spip.net)|151.80.20.125|:443... connected.
HTTP request sent, awaiting response ... 302 Found
Location: https://get.spip.net/spip_loader.php [following]
--2024-08-15 01:44:02--  https://get.spip.net/spip_loader.php
Resolving get.spip.net (get.spip.net)... 151.80.20.125
Connecting to get.spip.net (get.spip.net)|151.80.20.125|:443 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 265065 (259K) [application/octet-stream]
Saving to: 'spip_loader.php'

spip_loader.php              100%[=============================================>] 258.85K   419KB/s    in 0.6s

2024-08-15 01:44:04 (419 KB/s) - 'spip_loader.php' saved [265065/265065]
```

- Moved the *spip_loader.php* file to */var/www/html* and verify the file has been moved successfully



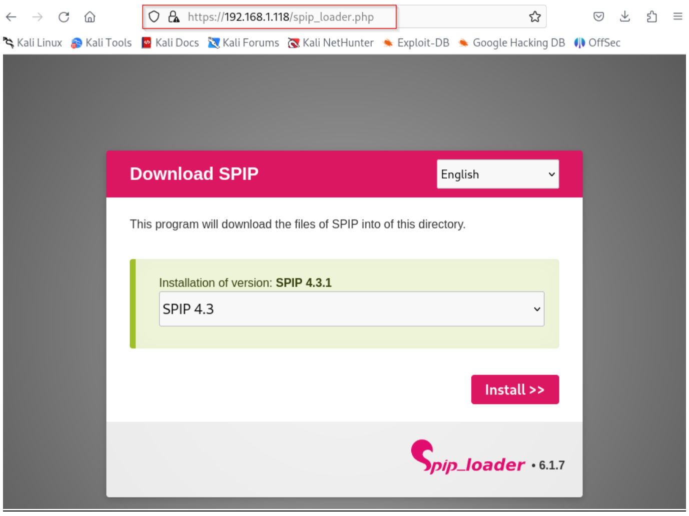
```
root@publisher:/home/newuser1$ sudo mv /home/newuser1/spip_loader.php /var/www/html/
root@publisher:/home/newuser1$
root@publisher:/home/newuser1$ ls /var/www/html/
total 268K
drwxrwx--- 2 www-data www-data 4.0K Aug 15 01:47 .
drwxr-xr-x 3 root     root     4.0K Jul 30  2023 ..
-rw-r--r-- 1 root     root     259K Aug  4 09:34 spip_loader.php
```

- *Giving correct permissions to the spip_loader.php*



```
root@publisher:/home/newuser1$ sudo chown www-data:www-data /var/www/html/spip_loader.php
root@publisher:/home/newuser1$
root@publisher:/home/newuser1$ sudo chmod 755 /var/www/html/spip_loader.php
```

- This shows that SPIP is ready to update to version 4.3.1



Download SPIP

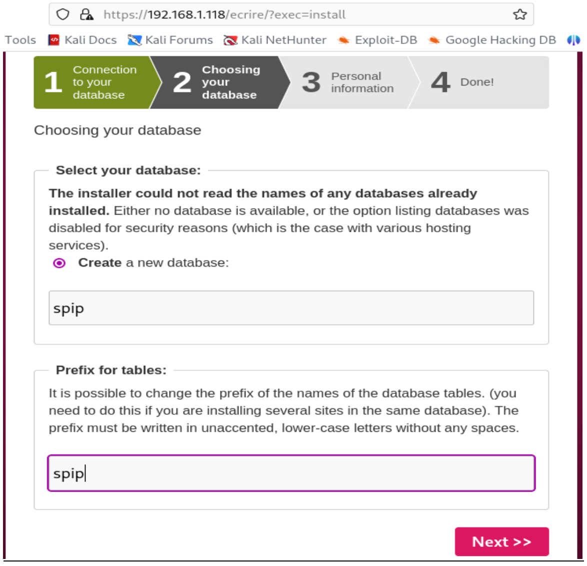
English

This program will download the files of SPIP into of this directory.

Installation of version: **SPIP 4.3.1**

SPIP 4.3

- Here the installer is suggesting to create a new database namped spip.
- The installation suggests building a new database called spip. If we do not have an existing SPIP database to connect to, you can use this option.
- If this is an update and we wish to use an existing database, make sure we are logged in as a user with the appropriate permissions to see the databases.
- We select next.



Choosing your database

Select your database:

The installer could not read the names of any databases already installed. Either no database is available, or the option listing databases was disabled for security reasons (which is the case with various hosting services).

● **Create** a new database:

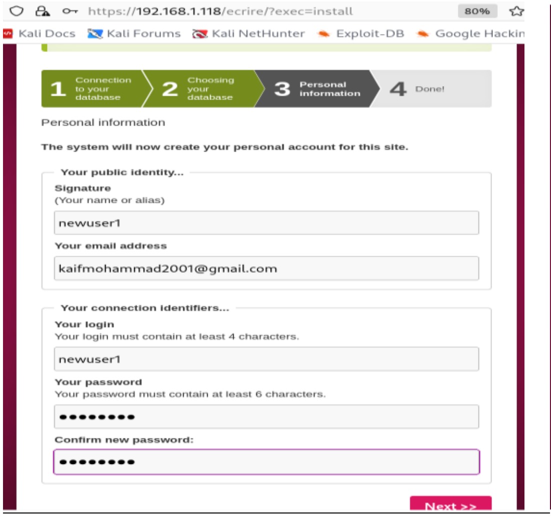
spip

Prefix for tables:

It is possible to change the prefix of the names of the database tables. (you need to do this if you are installing several sites in the same database). The prefix must be written in unaccented, lower-case letters without any spaces.

spip

- We gave credentials for personal account for SPIPS site.



Personal information

The system will now create your personal account for this site.

Your public identity...

Signature
(Your name or alias)

newuser1

Your email address

kaifmohammad2001@gmail.com

Your connection identifiers...

Your login
Your login must contain at least 4 characters.

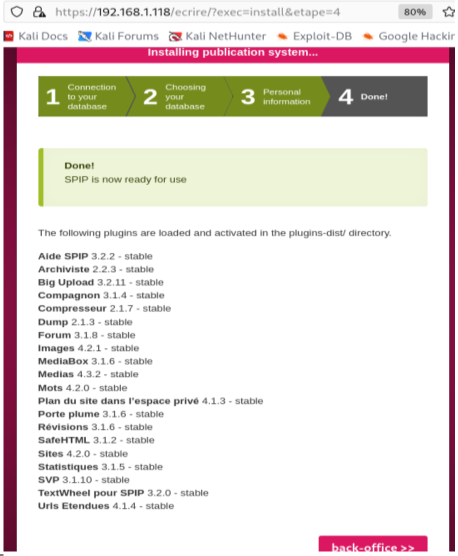
newuser1

Your password
Your password must contain at least 6 characters.

●●●●●●●●

Confirm new password:

●●●●●●●●

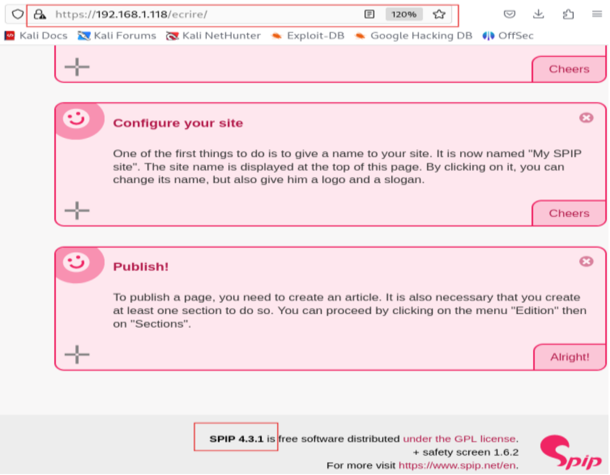


Done!
SPIP is now ready for use

The following plugins are loaded and activated in the plugins-dist/ directory.

Aide SPIP 3.2.2 - stable
Archiviste 2.2.3 - stable
Big Upload 3.2.11 - stable
Compagnon 3.1.4 - stable
Compresseur 2.1.7 - stable
Dump 2.1.3 - stable
Forum 3.1.8 - stable
Images 4.2.1 - stable
MediaBox 3.1.6 - stable
Medias 4.3.2 - stable
Mots 4.2.0 - stable
Plan du site dans l'espace privé 4.1.3 - stable
Porte plume 3.1.6 - stable
Révisions 3.1.6 - stable
SafeHTML 3.1.2 - stable
Sites 4.2.0 - stable
Statistiques 3.1.5 - stable
SVP 3.1.10 - stable
TextWheel pour SPIP 3.2.0 - stable
Urls Etendues 4.1.4 - stable

- Here we can confirm that the SPIP version upgrade was successful. The version number is now 4.3.1



Configure your site

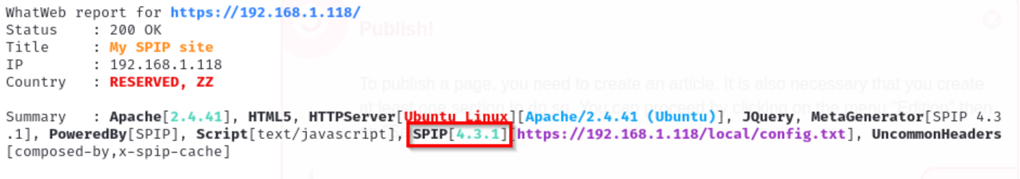
One of the first things to do is to give a name to your site. It is now named "My SPIP site". The site name is displayed at the top of this page. By clicking on it, you can change its name, but also give him a logo and a slogan.

Publish!

To publish a page, you need to create an article. It is also necessary that you create at least one section to do so. You can proceed by clicking on the menu "Edition" then on "Sections".

SPIP 4.3.1 is free software distributed under the GPL license.
+ safety screen 1.6.2
For more visit https://www.spip.net/en.

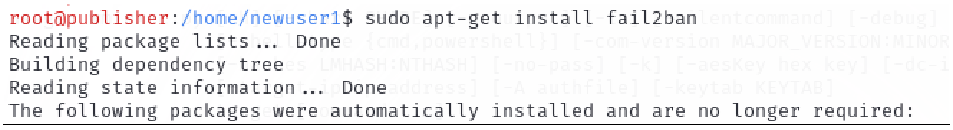
- And from the whatweb results we can confirm the SPIP 4.3.1



```
WhatWeb report for https://192.168.1.118/
Status     : 200 OK
Title      : My SPIP site
IP         : 192.168.1.118
Country    : RESERVED, ZZ

Summary    : Apache[2.4.41], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.41 (Ubuntu)], JQuery, MetaGenerator[SPIP 4.3
.1], PoweredBy[SPIP], Script[text/javascript], SPIP[4.3.1][https://192.168.1.118/local/config.txt], UncommonHeaders
[composed-by,x-spip-cache]
```

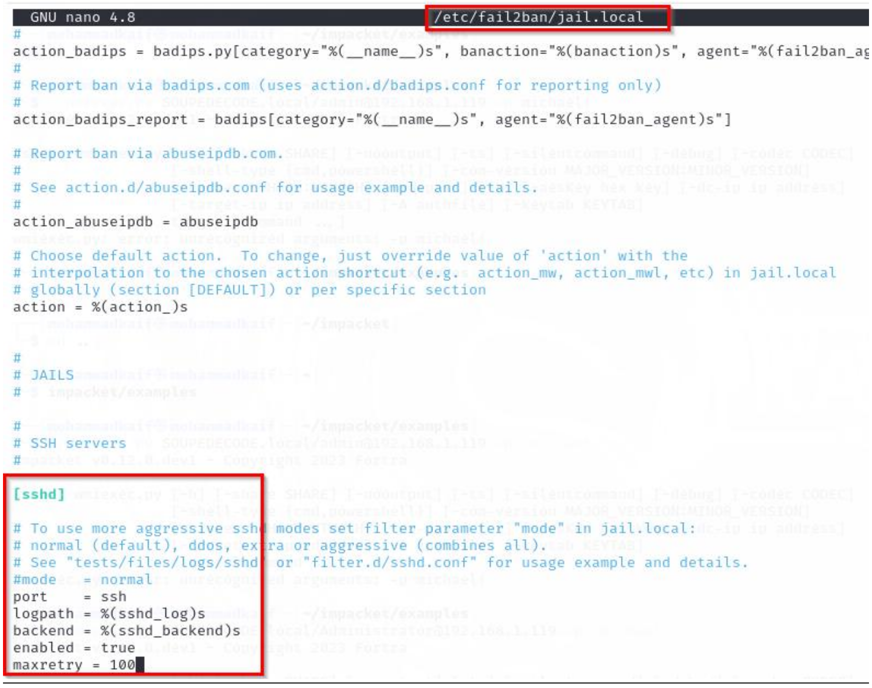
5. <u>Preventing brtute force on SSH</u>

- Systems are always prone to brute force attacks. Also, there is a lot of brute-for tools in the market, and we used some of them in the exploitation face. In order to preven publisher from brute force on SSH
- We installed Fail2Ban



```
root@publisher:/home/newuser1$ sudo apt-get install fail2ban
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
```

- We configured the file by setting the maxx entry to 100, this this is the number of failed login allowed before the IP is banned



```
  GNU nano 4.8                                       /etc/fail2ban/jail.local
#
action_badips = badips.py[category="%(__name__)s", banaction="%(banaction)s", agent="%(fail2ban_ag
#
# Report ban via badips.com (uses action.d/badips.conf for reporting only)
#
action_badips_report = badips[category="%(__name__)s", agent="%(fail2ban_agent)s"]

# Report ban via abuseipdb.com.
#
# See action.d/abuseipdb.conf for usage example and details.
#
action_abuseipdb = abuseipdb

# Choose default action.  To change, just override value of 'action' with the
# interpolation to the chosen action shortcut (e.g.  action_mw, action_mwl, etc) in jail.local
# globally (section [DEFAULT]) or per specific section
action = %(action_)s


#
# JAILS
#

#
# SSH servers
#

[sshd]

# To use more aggressive sshd modes set filter parameter "mode" in jail.local:
# normal (default), ddos, extra or aggressive (combines all).
# See "tests/files/logs/sshd" or "filter.d/sshd.conf" for usage example and details.
#mode   = normal
port    = ssh
logpath = %(sshd_log)s
backend = %(sshd_backend)s
enabled = true
maxretry = 100
```

s.