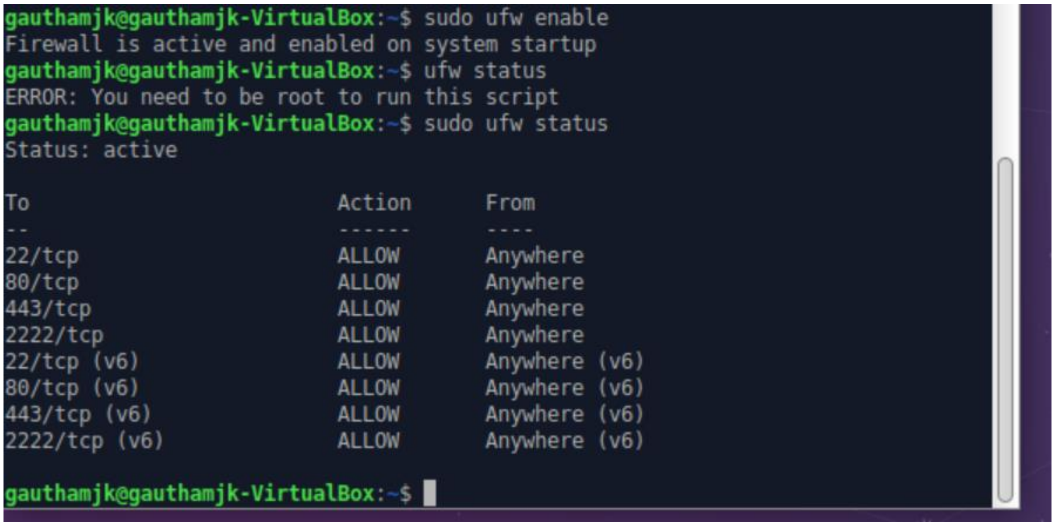
ITIS 5246: Firewalls

Gautham Krishna Kumar

In this assignment, we build a firewall policy. The assignment has been divided into 2 parts.

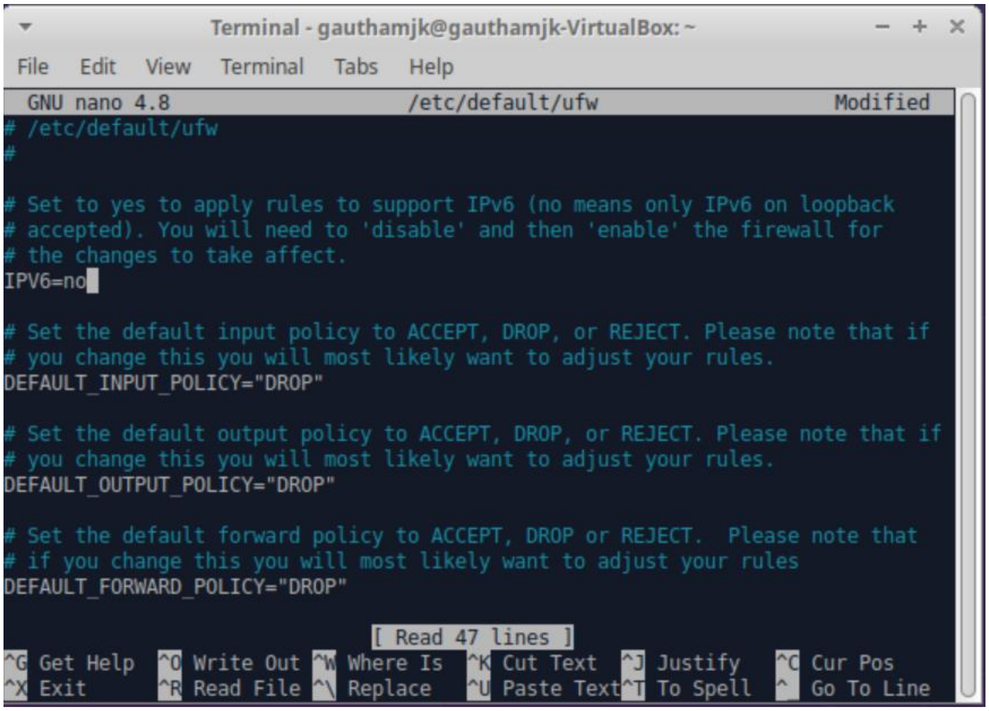
- First, we use UFW to develop a firewall policy to allow access to HTTP, HTTPS, SSH and Port 2222. All other ports need to be rejected.
- To view that only those 4 ports are accepted is by using the command: sudo ufw status



```
gauthamjk@gauthamjk-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
gauthamjk@gauthamjk-VirtualBox:~$ ufw status
ERROR: You need to be root to run this script
gauthamjk@gauthamjk-VirtualBox:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
443/tcp                    ALLOW       Anywhere
2222/tcp                   ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)
443/tcp (v6)               ALLOW       Anywhere (v6)
2222/tcp (v6)              ALLOW       Anywhere (v6)

gauthamjk@gauthamjk-VirtualBox:~$
```

- To allow outgoing traffic, we use sudo ufw allow outgoing and for denying incoming traffic, we use sudo ufw deny incoming.

- Now, we need to edit the /etc/default/ufw and change the IPV6 to No.



```
# /etc/default/ufw
#

# Set to yes to apply rules to support IPv6 (no means only IPv6 on loopback
# accepted). You will need to 'disable' and then 'enable' the firewall for
# the changes to take affect.
IPV6=no

# Set the default input policy to ACCEPT, DROP, or REJECT. Please note that if
# you change this you will most likely want to adjust your rules.
DEFAULT_INPUT_POLICY="DROP"

# Set the default output policy to ACCEPT, DROP, or REJECT. Please note that if
# you change this you will most likely want to adjust your rules.
DEFAULT_OUTPUT_POLICY="DROP"

# Set the default forward policy to ACCEPT, DROP or REJECT.  Please note that
# if you change this you will most likely want to adjust your rules
DEFAULT_FORWARD_POLICY="DROP"
```

- To redirect HTTP and HTTPS traffic to the server container, we use the following lxc commands:

```
• lxc config device add server nginx_http proxy listen=tcp:0.0.0.0:80 connect=tc
  p:127.0.0.1:80

• lxc config device add server nginx_https proxy listen=tcp:0.0.0.0:443 connect=
  tcp:127.0.0.1:443
```

- To redirect SSH traffic to the server container, we use the following lxc command:

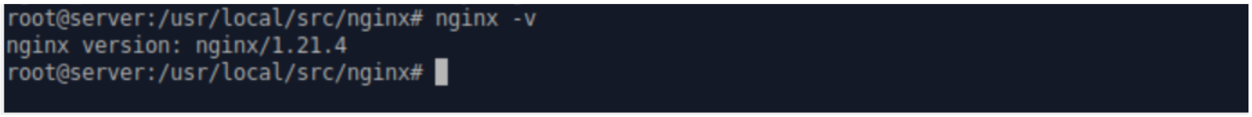
```
• lxc config device add server nginx_ssh proxy listen=tcp:0.0.0.0:22 connect=tcp
  :127.0.0.1:22
```

- To redirect port 2222 to the client container’s SSH service, we use the following command:

```
• lxc config device add client nginx_2222 proxy listen=tcp:0.0.0.0:2222 connect=
  tcp:127.0.0.1:2222
```

- Next, we need to get inside the server container.

- First, we need to make sure that we have the latest nginx version running on the server container by typing nginx -v.



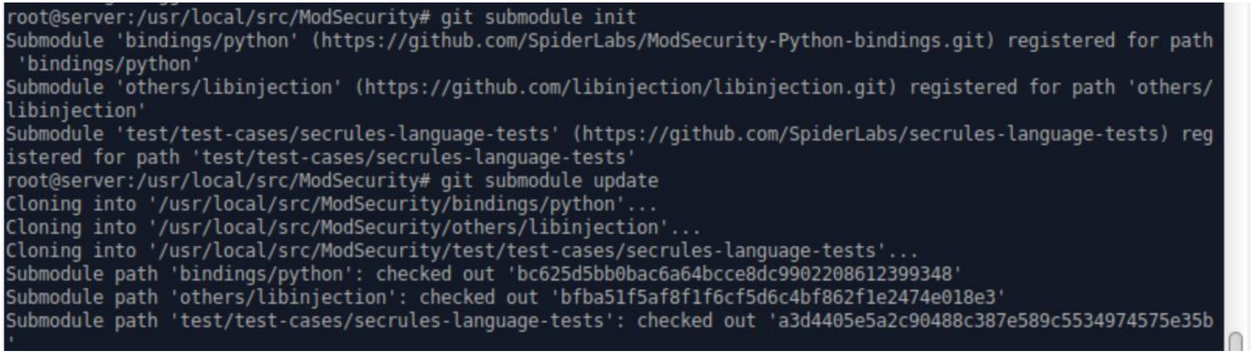
```
root@server:/usr/local/src/nginx# nginx -v
nginx version: nginx/1.21.4
root@server:/usr/local/src/nginx# 
```

- To install ModSecurity, we need to clone the libmodsecurity3 git repository onto the server container by using the following command:

```
git clone --depth 1 -b v3/master --single-branch
https://github.com/SpiderLabs/ModSecurity
/usr/local/src/ModSecurity/
```

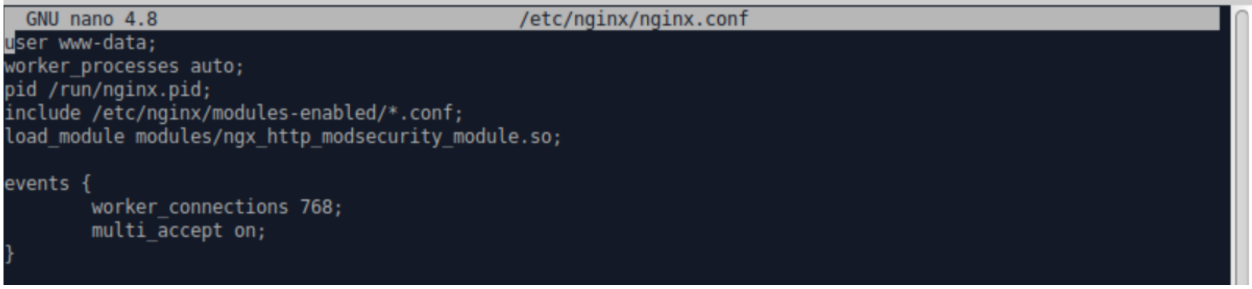
- To install the ModSecurity dependencies, we install the following git submodules:

```
git submodule init
git submodule update
```

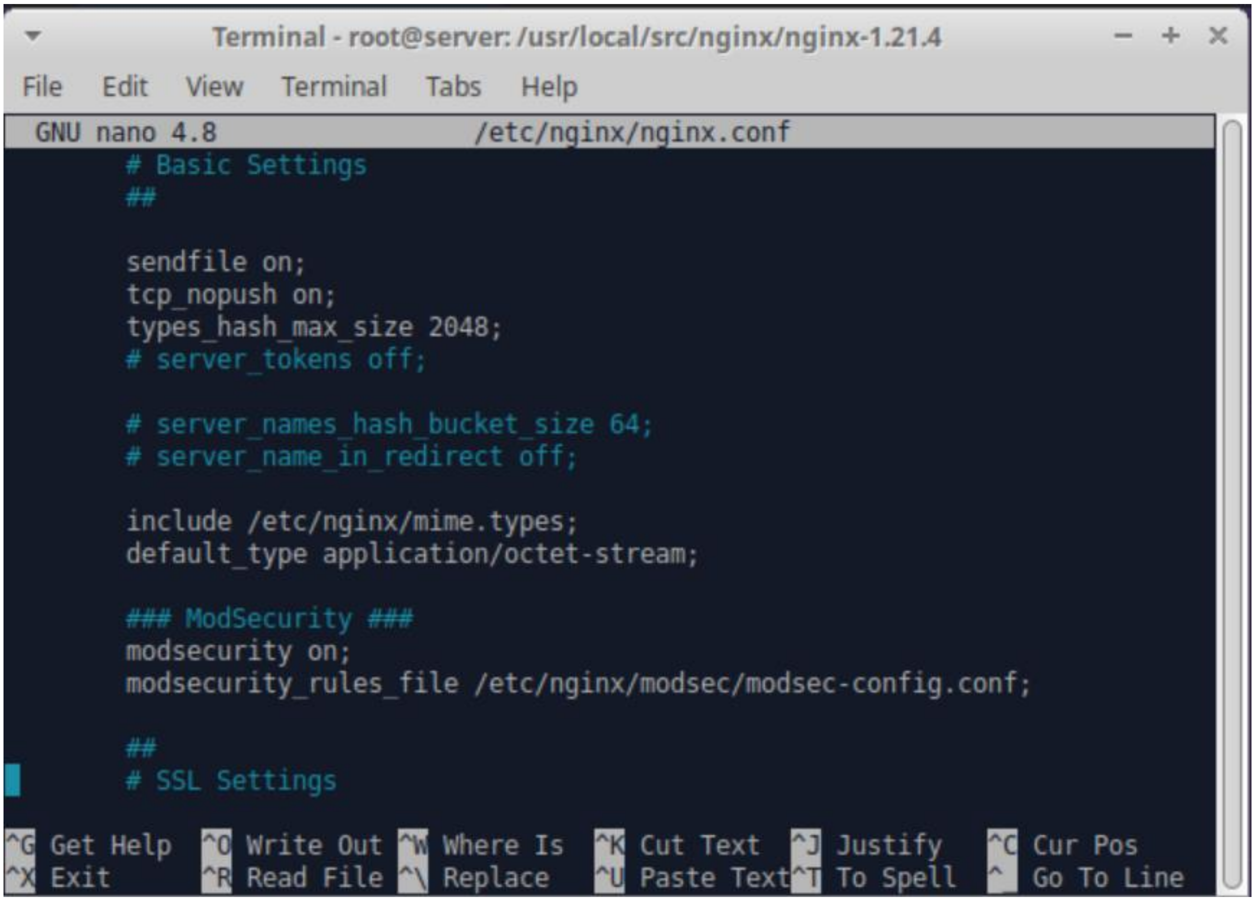


```
root@server:/usr/local/src/ModSecurity# git submodule init
Submodule 'bindings/python' (https://github.com/SpiderLabs/ModSecurity-Python-bindings.git) registered for path
 'bindings/python'
Submodule 'others/libinjection' (https://github.com/libinjection/libinjection.git) registered for path 'others/
libinjection'
Submodule 'test/test-cases/secrules-language-tests' (https://github.com/SpiderLabs/secrules-language-tests) reg
istered for path 'test/test-cases/secrules-language-tests'
root@server:/usr/local/src/ModSecurity# git submodule update
Cloning into '/usr/local/src/ModSecurity/bindings/python'...
Cloning into '/usr/local/src/ModSecurity/others/libinjection'...
Cloning into '/usr/local/src/ModSecurity/test/test-cases/secrules-language-tests'...
Submodule path 'bindings/python': checked out 'bc625d5bb0bac6a64bcce8dc9902208612399348'
Submodule path 'others/libinjection': checked out 'bfba51f5af8f1f6cf5d6c4bf862f1e2474e018e3'
Submodule path 'test/test-cases/secrules-language-tests': checked out 'a3d4405e5a2c90488c387e589c5534974575e35b
'
```

- We need to enable ModSecurity in nginx.conf:

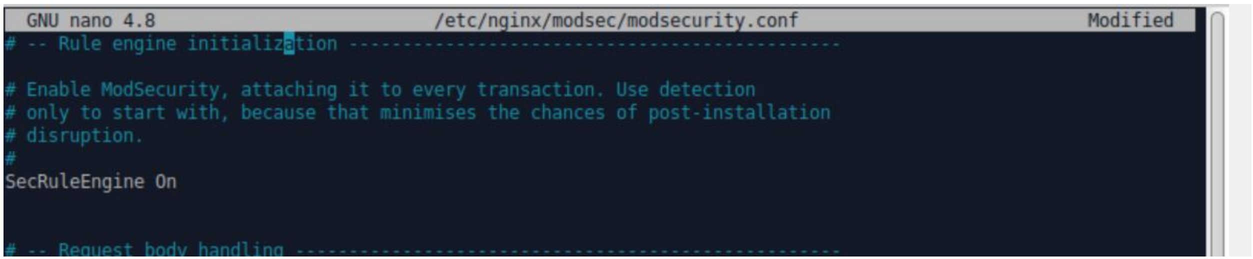


```
  GNU nano 4.8                              /etc/nginx/nginx.conf
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;
load_module modules/ngx_http_modsecurity_module.so;

events {
        worker_connections 768;
        multi_accept on;
}
```

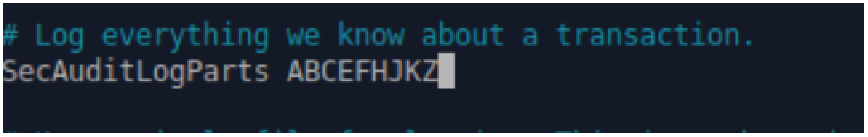


```
        # Basic Settings
        ##

        sendfile on;
        tcp_nopush on;
        types_hash_max_size 2048;
        # server_tokens off;

        # server_names_hash_bucket_size 64;
        # server_name_in_redirect off;

        include /etc/nginx/mime.types;
        default_type application/octet-stream;

        ### ModSecurity ###
        modsecurity on;
        modsecurity_rules_file /etc/nginx/modsec/modsec-config.conf;

        ##
        # SSL Settings
```

- We make the necessary changes in modsecurity.conf:

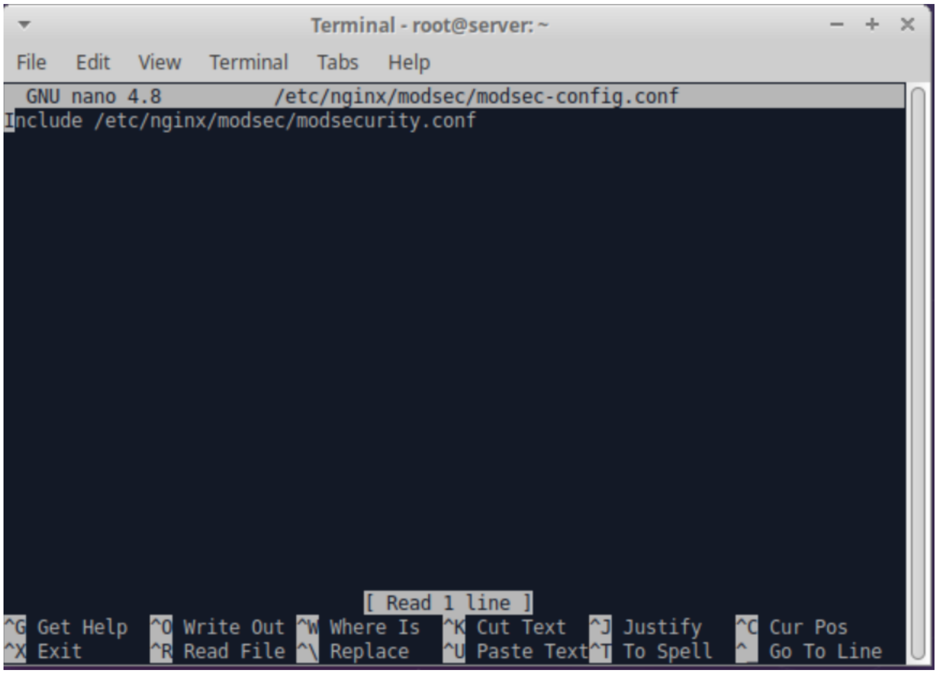


```
# -- Rule engine initialization ----------------------------------------------

# Enable ModSecurity, attaching it to every transaction. Use detection
# only to start with, because that minimises the chances of post-installation
# disruption.
#
SecRuleEngine On


# -- Request body handling ---------------------------------------------------
```



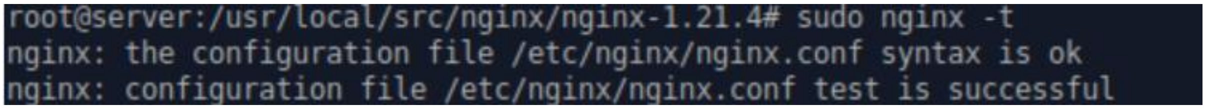
```
# Log everything we know about a transaction.
SecAuditLogParts ABCEFHJKZ
```

- We include the modsecurity.conf in the modsec-config.conf file.


Terminal - root@server: ~
File Edit View Terminal Tabs Help
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line

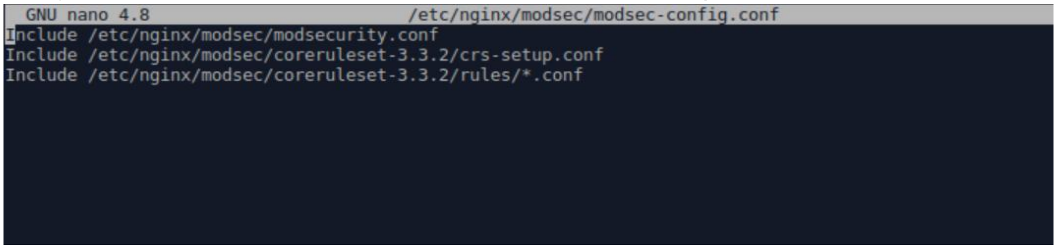
```
  GNU nano 4.8            /etc/nginx/modsec/modsec-config.conf
Include /etc/nginx/modsec/modsecurity.conf

                              [ Read 1 line ]
```

- We perform a dry run on the Nginx service to ensure that it is running, we use the command: sudo nginx -t.



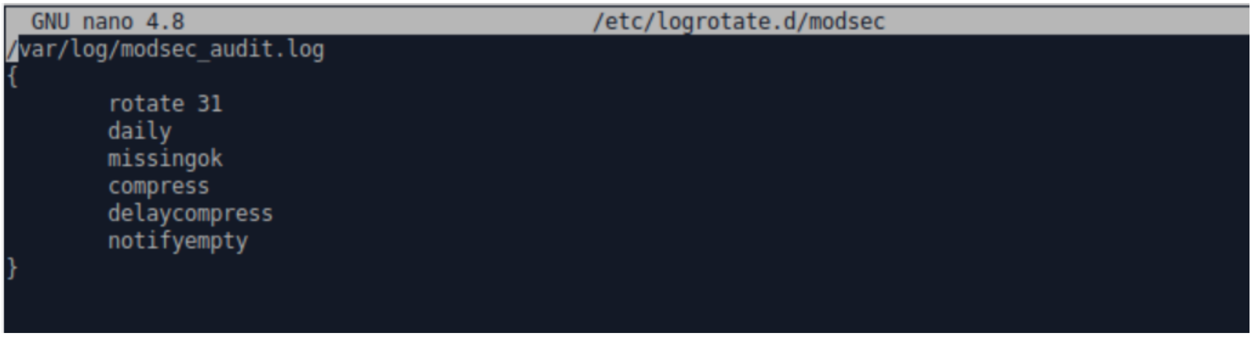
```
root@server:/usr/local/src/nginx/nginx-1.21.4# sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

- With ModSecurity installed, the next thing we need to install is OWASP Core Rule Set for ModSecurity. We need to download the OWASP CRS v3.3.2 zip file in our server container.
- Unzip the file and add two more rules to the modsec-config.conf file:



```
  GNU nano 4.8                        /etc/nginx/modsec/modsec-config.conf
Include /etc/nginx/modsec/modsecurity.conf
Include /etc/nginx/modsec/coreruleset-3.3.2/crs-setup.conf
Include /etc/nginx/modsec/coreruleset-3.3.2/rules/*.conf
```

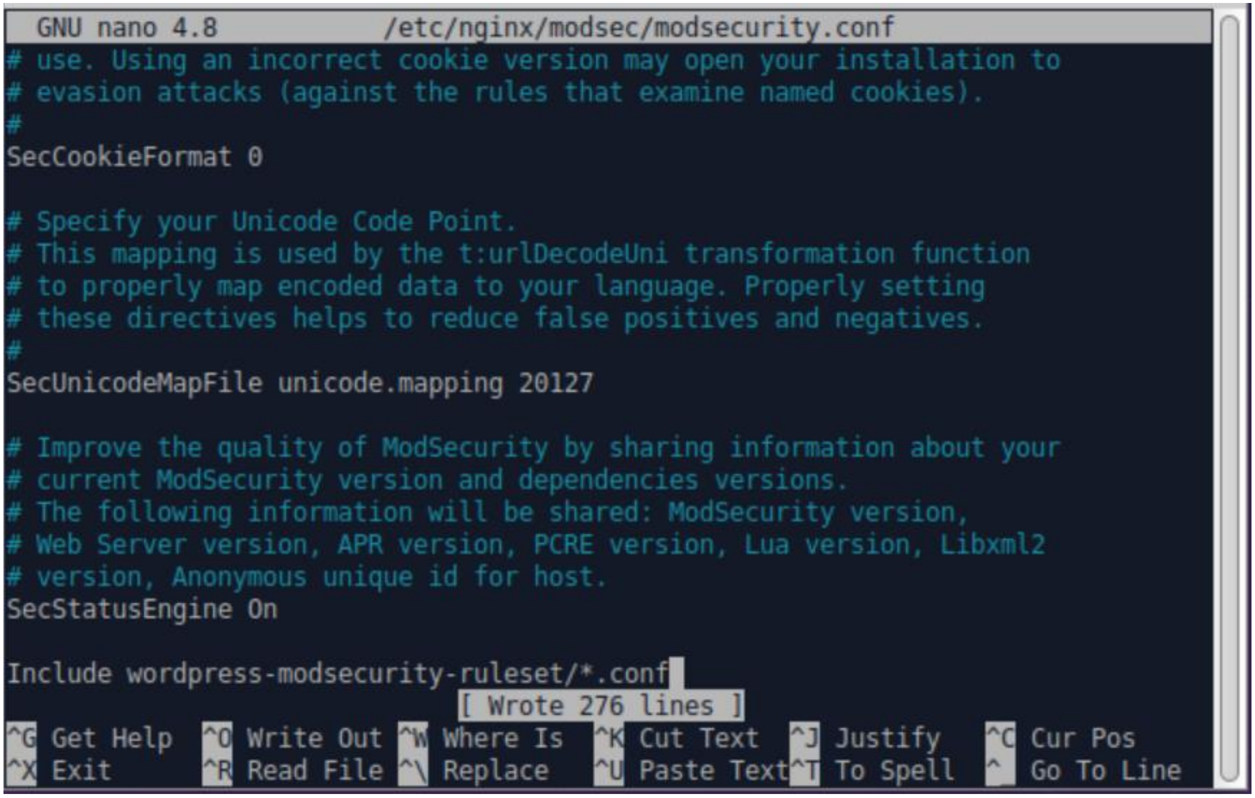
- Once again, we need to test nginx to ensure that there are no errors.
- Created a ModSecurity LogRotate file to monitor the logs.



```
  GNU nano 4.8                              /etc/logrotate.d/modsec
/var/log/modsec_audit.log
{
        rotate 31
        daily
        missingok
        compress
        delaycompress
        notifyempty
}
```

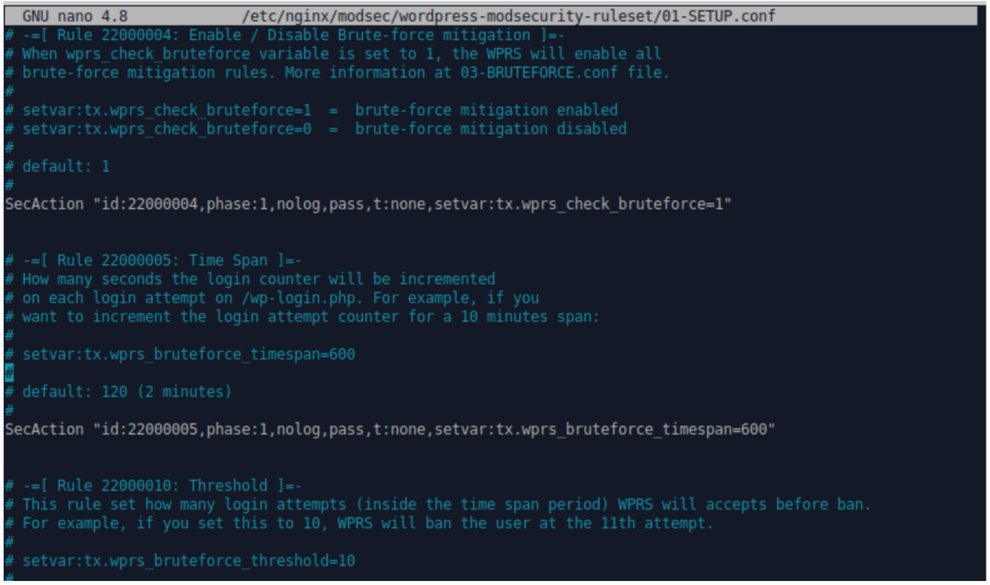
- Now, we need to download WordPress ModSecurity RuleSet. We clone the wordpress-modsecurity-ruleset.git into the modsec folder.
- To install WordPress in OWASP CRS, we need to add the following at the end of the modsecurity.conf file:

```
Include wordpress-modsecurity-ruleset/*.conf
```

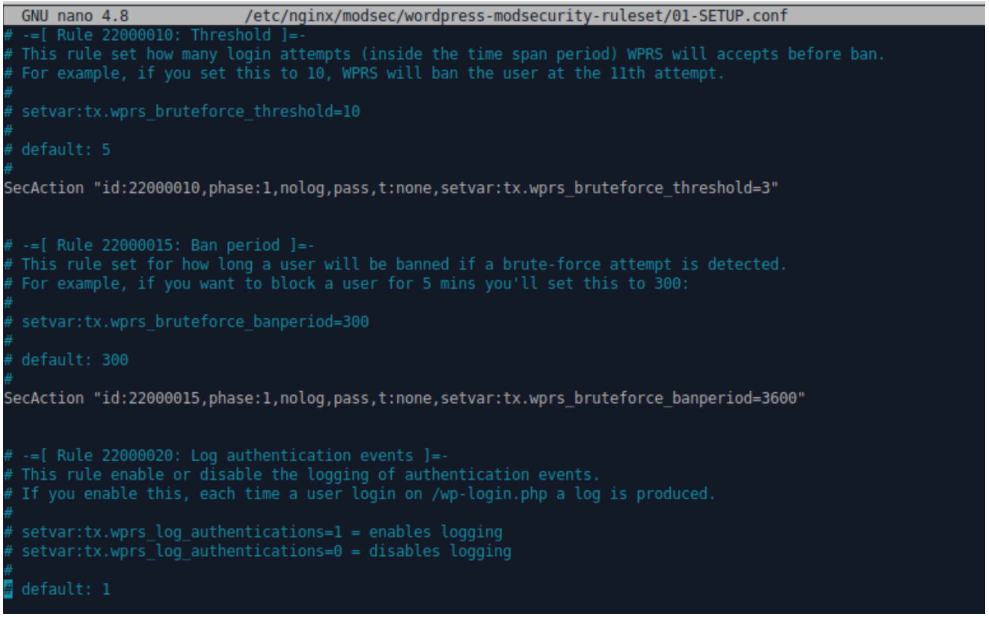


```
  GNU nano 4.8              /etc/nginx/modsec/modsecurity.conf
# use. Using an incorrect cookie version may open your installation to
# evasion attacks (against the rules that examine named cookies).
#
SecCookieFormat 0

# Specify your Unicode Code Point.
# This mapping is used by the t:urlDecodeUni transformation function
# to properly map encoded data to your language. Properly setting
# these directives helps to reduce false positives and negatives.
#
SecUnicodeMapFile unicode.mapping 20127

# Improve the quality of ModSecurity by sharing information about your
# current ModSecurity version and dependencies versions.
# The following information will be shared: ModSecurity version,
# Web Server version, APR version, PCRE version, Lua version, Libxml2
# version, Anonymous unique id for host.
SecStatusEngine On

Include wordpress-modsecurity-ruleset/*.conf
                              [ Wrote 276 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

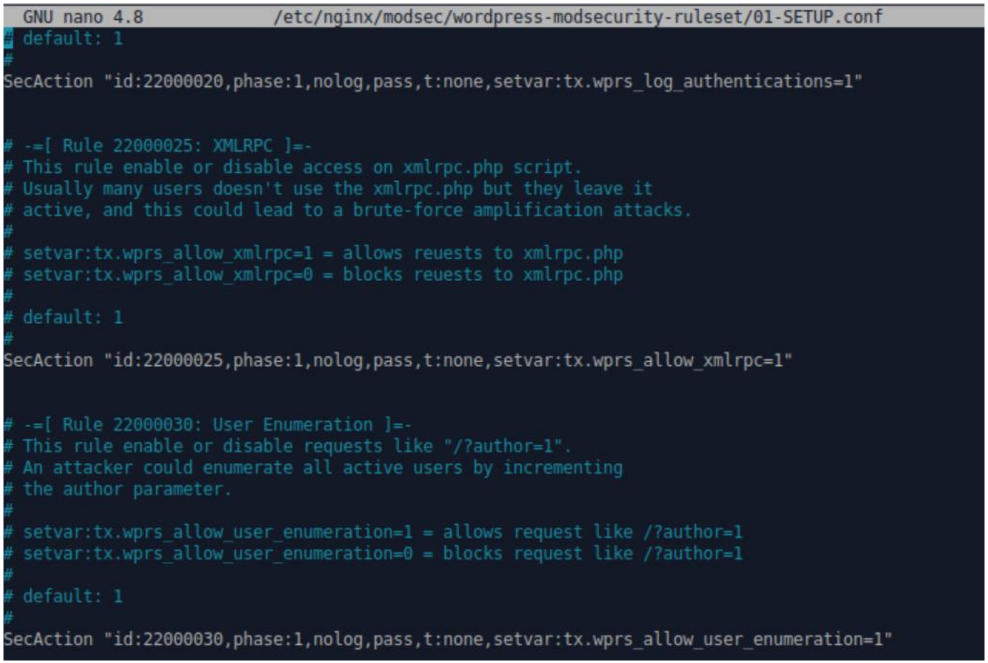
- To configure WPRS, we need to uncomment the rules in the 01-SETUP.conf file.



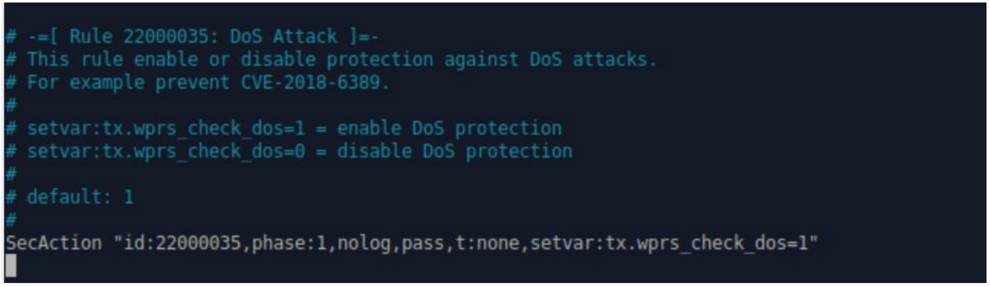
```
  GNU nano 4.8              /etc/nginx/modsec/wordpress-modsecurity-ruleset/01-SETUP.conf
# -=[ Rule 22000004: Enable / Disable Brute-force mitigation ]=-
# When wprs_check_bruteforce variable is set to 1, the WPRS will enable all
# brute-force mitigation rules. More information at 03-BRUTEFORCE.conf file.
#
# setvar:tx.wprs_check_bruteforce=1  =  brute-force mitigation enabled
# setvar:tx.wprs_check_bruteforce=0  =  brute-force mitigation disabled
#
# default: 1
#
SecAction "id:22000004,phase:1,nolog,pass,t:none,setvar:tx.wprs_check_bruteforce=1"


# -=[ Rule 22000005: Time Span ]=-
# How many seconds the login counter will be incremented
# on each login attempt on /wp-login.php. For example, if you
# want to increment the login attempt counter for a 10 minutes span:
#
# setvar:tx.wprs_bruteforce_timespan=600
#
# default: 120 (2 minutes)
#
SecAction "id:22000005,phase:1,nolog,pass,t:none,setvar:tx.wprs_bruteforce_timespan=600"


# -=[ Rule 22000010: Threshold ]=-
# This rule set how many login attempts (inside the time span period) WPRS will accepts before ban.
# For example, if you set this to 10, WPRS will ban the user at the 11th attempt.
#
# setvar:tx.wprs_bruteforce_threshold=10
#
```



```
  GNU nano 4.8              /etc/nginx/modsec/wordpress-modsecurity-ruleset/01-SETUP.conf
# -=[ Rule 22000010: Threshold ]=-
# This rule set how many login attempts (inside the time span period) WPRS will accepts before ban.
# For example, if you set this to 10, WPRS will ban the user at the 11th attempt.
#
# setvar:tx.wprs_bruteforce_threshold=10
#
# default: 5
#
SecAction "id:22000010,phase:1,nolog,pass,t:none,setvar:tx.wprs_bruteforce_threshold=3"


# -=[ Rule 22000015: Ban period ]=-
# This rule set for how long a user will be banned if a brute-force attempt is detected.
# For example, if you want to block a user for 5 mins you'll set this to 300:
#
# setvar:tx.wprs_bruteforce_banperiod=300
#
# default: 300
#
SecAction "id:22000015,phase:1,nolog,pass,t:none,setvar:tx.wprs_bruteforce_banperiod=3600"


# -=[ Rule 22000020: Log authentication events ]=-
# This rule enable or disable the logging of authentication events.
# If you enable this, each time a user login on /wp-login.php a log is produced.
#
# setvar:tx.wprs_log_authentications=1 = enables logging
# setvar:tx.wprs_log_authentications=0 = disables logging
#
# default: 1
```

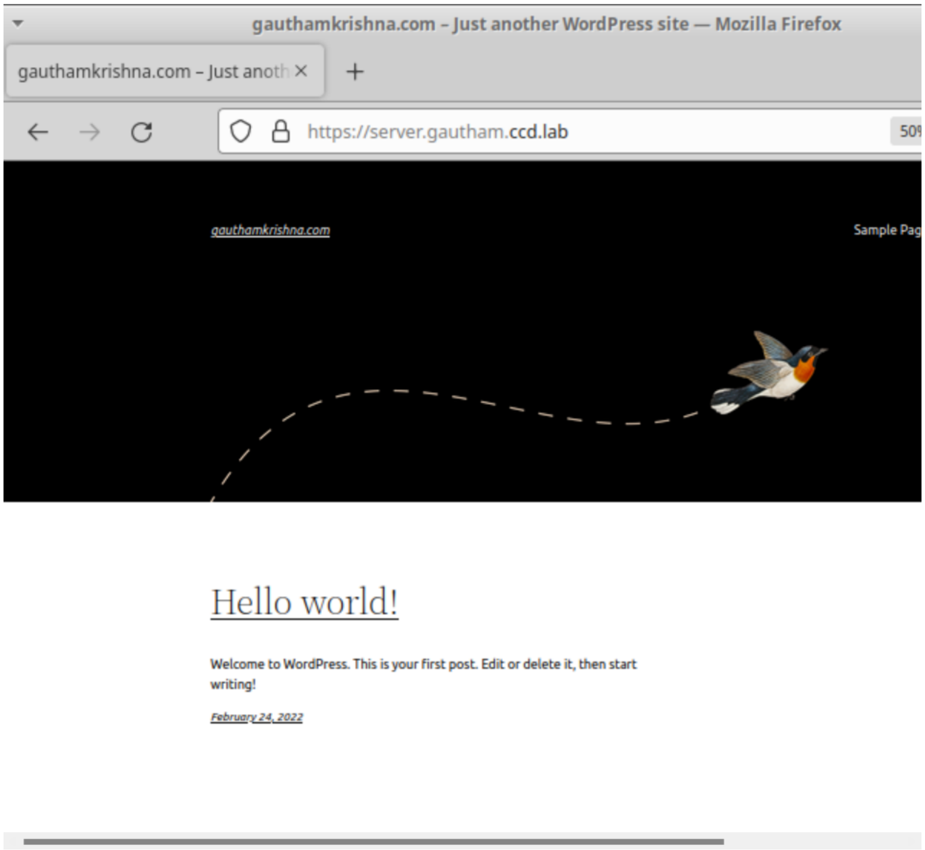


```
  GNU nano 4.8              /etc/nginx/modsec/wordpress-modsecurity-ruleset/01-SETUP.conf
# default: 1
#
SecAction "id:22000020,phase:1,nolog,pass,t:none,setvar:tx.wprs_log_authentications=1"


# -=[ Rule 22000025: XMLRPC ]=-
# This rule enable or disable access on xmlrpc.php script.
# Usually many users doesn't use the xmlrpc.php but they leave it
# active, and this could lead to a brute-force amplification attacks.
#
# setvar:tx.wprs_allow_xmlrpc=1 = allows reuests to xmlrpc.php
# setvar:tx.wprs_allow_xmlrpc=0 = blocks reuests to xmlrpc.php
#
# default: 1
#
SecAction "id:22000025,phase:1,nolog,pass,t:none,setvar:tx.wprs_allow_xmlrpc=1"


# -=[ Rule 22000030: User Enumeration ]=-
# This rule enable or disable requests like "/?author=1".
# An attacker could enumerate all active users by incrementing
# the author parameter.
#
# setvar:tx.wprs_allow_user_enumeration=1 = allows request like /?author=1
# setvar:tx.wprs_allow_user_enumeration=0 = blocks request like /?author=1
#
# default: 1
#
SecAction "id:22000030,phase:1,nolog,pass,t:none,setvar:tx.wprs_allow_user_enumeration=1"
```



```
# -=[ Rule 22000035: DoS Attack ]=-
# This rule enable or disable protection against DoS attacks.
# For example prevent CVE-2018-6389.
#
# setvar:tx.wprs_check_dos=1 = enable DoS protection
# setvar:tx.wprs_check_dos=0 = disable DoS protection
#
# default: 1
#
SecAction "id:22000035,phase:1,nolog,pass,t:none,setvar:tx.wprs_check_dos=1"
```

- With all this installed, we need to test the web server to ensure that we have the wordpress screen up and running. To do that, we navigate to https://server.gautham.ccd.lab. If Wordpress is working, we can confirm that the HTTP and HTTPS requests have been redirected to the server container.

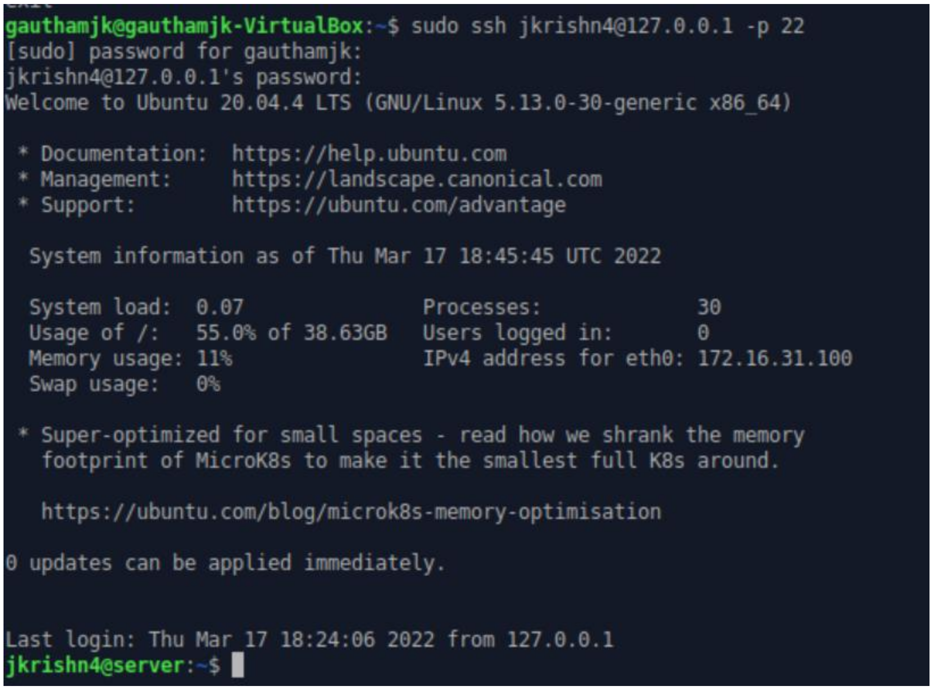

gauthamkrishna.com – Just another WordPress site — Mozilla Firefox
gauthamkrishna.com – Just anoth
https://server.gautham.ccd.lab
50
gauthamkrishna.com
Sample Pag

Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

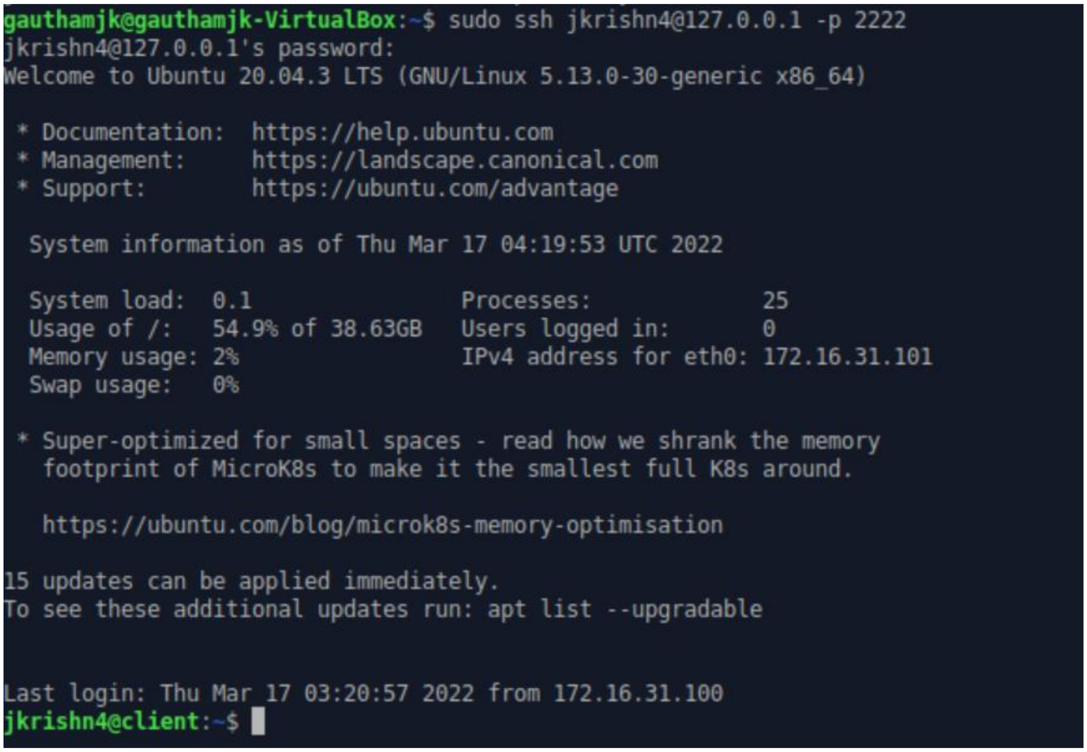
February 24, 2022

- Next, we need to ensure that the SSH requests are redirected to the server container, we use the command ssh jkrishn4@127.0.0.1 -p 22. This will connect us to the server container.



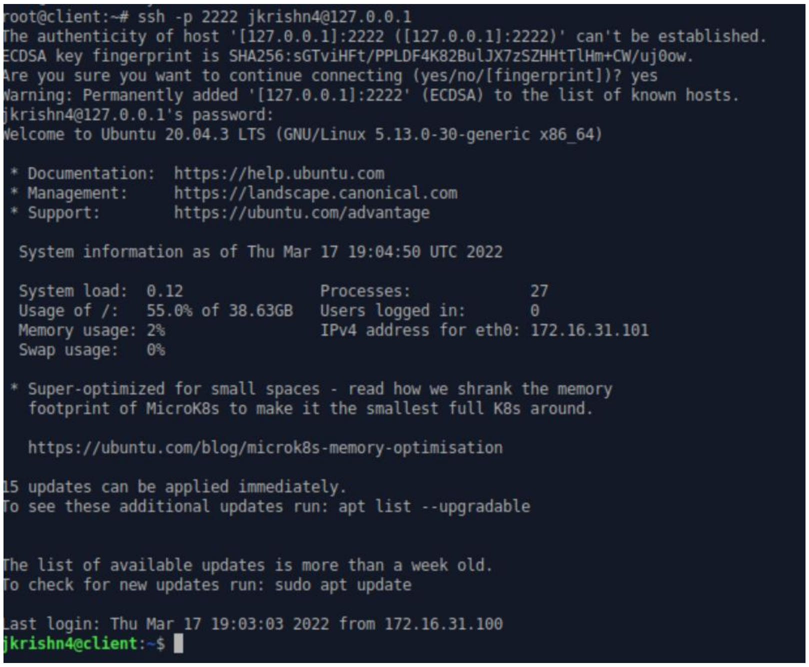
```
gauthamjk@gauthamjk-VirtualBox:~$ sudo ssh jkrishn4@127.0.0.1 -p 22
[sudo] password for gauthamjk:
jkrishn4@127.0.0.1's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.13.0-30-generic x86_64)

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

  System information as of Thu Mar 17 18:45:45 UTC 2022

  System load:  0.07              Processes:             30
  Usage of /:   55.0% of 38.63GB   Users logged in:       0
  Memory usage: 11%               IPv4 address for eth0: 172.16.31.100
  Swap usage:   0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

0 updates can be applied immediately.


Last login: Thu Mar 17 18:24:06 2022 from 127.0.0.1
jkrishn4@server:~$ 
```

- Lastly, to ensure that port 2222 is redirected to the client container we use the command: ssh jkrishn4@127.0.0.1 -p 2222. This will connect us to the client container.



```
gauthamjk@gauthamjk-VirtualBox:~$ sudo ssh jkrishn4@127.0.0.1 -p 2222
jkrishn4@127.0.0.1's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.13.0-30-generic x86_64)

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

  System information as of Thu Mar 17 04:19:53 UTC 2022

  System load:  0.1               Processes:             25
  Usage of /:   54.9% of 38.63GB   Users logged in:       0
  Memory usage: 2%                IPv4 address for eth0: 172.16.31.101
  Swap usage:   0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

15 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


Last login: Thu Mar 17 03:20:57 2022 from 172.16.31.100
jkrishn4@client:~$
```

We can also do this inside the client container.



```
root@client:~# ssh -p 2222 jkrishn4@127.0.0.1
The authenticity of host '[127.0.0.1]:2222 ([127.0.0.1]:2222)' can't be established.
ECDSA key fingerprint is SHA256:sGTviHFt/PPLDF4K82BulJX7zSZHHtTlHm+CW/uj0ow.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[127.0.0.1]:2222' (ECDSA) to the list of known hosts.
jkrishn4@127.0.0.1's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.13.0-30-generic x86_64)

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

  System information as of Thu Mar 17 19:04:50 UTC 2022

  System load:  0.12              Processes:             27
  Usage of /:   55.0% of 38.63GB   Users logged in:       0
  Memory usage: 2%                IPv4 address for eth0: 172.16.31.101
  Swap usage:   0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

15 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Thu Mar 17 19:03:03 2022 from 172.16.31.100
jkrishn4@client:~$
```

Conclusion:

We have successfully deployed a firewall using the specified rulesets.