<https://viblo.asia/p/huong-dan-cai-dat-nginx-luajit-bat-headerbody-cua-request-response-4P856LyaZY3>

**Hướng dẫn cài đặt Nginx + LuaJIT bắt Header/Body của Request-Response**

[tuanduong122](https://viblo.asia/tags/tuanduong122)[capture header](https://viblo.asia/tags/capture-header)[luajit](https://viblo.asia/tags/luajit)[nginx](https://viblo.asia/tags/nginx)[Lua](https://viblo.asia/tags/lua)

Hướng dẫn cài đặt Nginx + LuaJIT bắt Header #tuanduong122

Bài toán đặt ra của mình: Nhiều lúc mình ko biết thằng vẹo Internet nào nó chèn header bậy vào hệ thống. Hoặc gặp mấy cái API đắng được bàn giao - ông đối tác bảo em cũng mới nhận code, mà ko biết nó hoạt động như nào. Mình sẽ chèn nginx (đứng trước/đứng sau) để bắt xem Header, Body chúng nó hoạt động ra sao. Từ đó ông Sys sẽ hiểu hệ thống hoạt động hơn (khổ).



Mình xin đưa nguồn tham khảo trước, mình dựa vào các hướng dẫn để tối ưu vào cài đặt của mình.

1. https://cuongquach.com/cau-hinh-nginx-ghi-log-thong-tin-header-http.html #Cường Quách , có rất nhiều bài viết hay

2. Tham khảo cấu hình: https://www.hardill.me.uk/wordpress/2018/03/14/logging-requests-and-response-with-nginx/

3. Tham khảo cài đặt: https://linuxscriptshub.com/nginx-httpguard-to-block-cc-attack-centos-6-9/ #trang web nhiều cái hay :D

4. Tham khảo cài đặt: https://tarunlalwani.com/post/building-nginx-with-lua/

Version mình sử dụng:

- Nginx-1.17.10 (Bản stable mới nhất tính đến 04.2020)

- LuaJIT-2.0.5

- lua-nginx-module-0.10.13

- ngx\_devel\_kit-0.3.1rc1

- Áp dụng cho cả Centos6/7. Ở đây mình đang test trên bản 7 (CentOS 6 mình cũng cài đặt thành công)

**Bước 1: Cài đặt thư viện cơ bản hỗ trợ nginx:**

yum install wget readline-devel pcre pcre-devel openssl openssl-devel zlib zlib-devel gcc -y

**Bước 2: Cài đặt / Chuẩn bị LuaJIT**

- Tải LuaJIT về và giải nén

cd /opt

wget http://luajit.org/download/LuaJIT-2.0.5.tar.gz

tar zxf LuaJIT-2.0.5.tar.gz

cd LuaJIT-2.0.5

- Chèn export ở trên đầu, lưu file lại

vim Makefile

export PREFIX= /opt/luajit-2.0.5 (thực ra không cần bước này)

- Thực hiện Compile LuaJIT

make

make install

(ta có thư mục sau cài là : /usr/local/bin/luajit-2.0.5)

Set biến môi trường tạm thời

export LUAJIT\_LIB=/usr/local/lib

export LUAJIT\_INC=/usr/local/include/luajit-2.0

**Bước 3: Cài đặt nginx và các thư viện:**

.

.

.

cd /opt

wget http://nginx.org/download/nginx-1.17.10.tar.gz

wget https://github.com/openresty/lua-nginx-module/archive/v0.10.13.tar.gz

wget https://github.com/simplresty/ngx\_devel\_kit/archive/v0.3.1rc1.tar.gz

tar -xvzf nginx-1.17.10.tar.gz

tar -xvzf v0.10.13.tar.gz

tar -xvzf v0.3.1rc1.tar.gz

cd nginx-1.17.10

mkdir -p /opt/nginx

mkdir /opt/nginx/logs/

mkdir /run

- Thực hiện Compile Nginx, ở đây mình đặt Nginx ở thư mục OPT ưa thích (không phải /etc)

./configure --sbin-path=/usr/bin/nginx --prefix=/opt/nginx --conf-path=/opt/nginx/nginx.conf --error-log-path=/opt/nginx/logs/error.log --http-log-path=/opt/nginx/logs/access.log --with-pcre --pid-path=/var/run/nginx.pid --with-http\_ssl\_module --with-http\_realip\_module --with-http\_stub\_status\_module --with-http\_stub\_status\_module --with-http\_gzip\_static\_module --with-http\_realip\_module --with-ld-opt="-Wl,-rpath,/usr/local/lib" --add-module=/opt/lua-nginx-module-0.10.13 --add-module=/opt/ngx\_devel\_kit-0.3.1rc1

make

make install

**Tạo file khởi động cho nginx (init file)**

touch /etc/init.d/nginx

chmod 755 /etc/init.d/nginx

vim /etc/init.d/nginx

#!/bin/sh

#

# nginx - this script starts and stops the nginx daemon

#

# chkconfig: - 85 15

# description: Nginx is an HTTP(S) server, HTTP(S) reverse \

# proxy and IMAP/POP3 proxy server

# processname: nginx

# config: /etc/nginx/nginx.conf

# config: /etc/sysconfig/nginx

# pidfile: /var/run/nginx.pid

# Source function library.

. /etc/rc.d/init.d/functions

# Source networking configuration.

. /etc/sysconfig/network

# Check that networking is up.

[ "$NETWORKING" = "no" ] && exit 0

nginx="/usr/bin/nginx"

prog=$(basename $nginx)

NGINX\_CONF\_FILE="/opt/nginx/nginx.conf"

[ -f /etc/sysconfig/nginx ] && . /etc/sysconfig/nginx

lockfile=/var/lock/subsys/nginx

make\_dirs() {

# make required directories

user=`$nginx -V 2>&1 | grep "configure arguments:" | sed 's/[^\*]\*--user=\([^ ]\*\).\*/\1/g' -`

if [ -z "`grep $user /etc/passwd`" ]; then

useradd -M -s /bin/nologin $user

fi

options=`$nginx -V 2>&1 | grep 'configure arguments:'`

for opt in $options; do

if [ `echo $opt | grep '.\*-temp-path'` ]; then

value=`echo $opt | cut -d "=" -f 2`

if [ ! -d "$value" ]; then

# echo "creating" $value

mkdir -p $value && chown -R $user $value

fi

fi

done

}

start() {

[ -x $nginx ] || exit 5

[ -f $NGINX\_CONF\_FILE ] || exit 6

# make\_dirs

echo -n $"Starting $prog: "

daemon $nginx -c $NGINX\_CONF\_FILE

retval=$?

echo

[ $retval -eq 0 ] && touch $lockfile

return $retval

}

stop() {

echo -n $"Stopping $prog: "

killproc $prog -QUIT

retval=$?

echo

[ $retval -eq 0 ] && rm -f $lockfile

return $retval

}

restart() {

configtest || return $?

stop

sleep 1

start

}

reload() {

configtest || return $?

echo -n $"Reloading $prog: "

killproc $nginx -HUP

RETVAL=$?

echo

}

force\_reload() {

restart

}

configtest() {

$nginx -t -c $NGINX\_CONF\_FILE

}

rh\_status() {

status $prog

}

rh\_status\_q() {

rh\_status >/dev/null 2>&1

}

case "$1" in

start)

rh\_status\_q && exit 0

$1

;;

stop)

rh\_status\_q || exit 0

$1

;;

restart|configtest)

$1

;;

reload)

rh\_status\_q || exit 7

$1

;;

force-reload)

force\_reload

;;

status)

rh\_status

;;

condrestart|try-restart)

rh\_status\_q || exit 0

;;

\*)

echo $"Usage: $0 {start|stop|status|restart|condrestart|try-restart|reload|force-reload|configtest}"

exit 2

esac

**Test nginx syntax**

[root@localhost setup]# nginx -t

nginx: the configuration file /opt/nginx/nginx.conf syntax is ok

nginx: configuration file /opt/nginx/nginx.conf test is successful

#>>>>>>> NGON, Tạm ổn

**Add nginx vào systemctl và restart**

/etc/init.d/nginx status

Sẽ lỗi: Unit nginx.service could not be found.

--

systemctl daemon-reload

/etc/init.d/nginx restart

**Dọn rác khi cài:**

mkdir /opt/setup

cd /opt

mv LuaJIT-2.0.5 /opt/setup/

mv LuaJIT-2.0.5.tar.gz /opt/setup/

mv lua-nginx-module-0.10.13 /opt/setup/

mv nginx-1.17.10 /opt/setup/

mv nginx-1.17.10.tar.gz /opt/setup/

mv ngx\_devel\_kit-0.3.1rc1 /opt/setup/

mv v0.10.13.tar.gz /opt/setup/

mv v0.3.1rc1.tar.gz /opt/setup/

**Bước4 : Cấu hình test LUA**

(Có thể bỏ qua bước này và đi luôn vào BƯỚC 5 - ở đây mình muốn test lua hoạt động oke chưa)

Thêm location này vào nginx (có thể bỏ qua bước này và copy nguyên file config của mình ở bước dưới - BƯỚC 5):

vim /opt/nginx/nginx.conf

location /sum {

content\_by\_lua\_block {

local args = ngx.req.get\_uri\_args();

ngx.say(args.a + args.b)

}

}

Test syntax nginx và reload lại

[root@localhost opt]# nginx -t

nginx: the configuration file /opt/nginx/nginx.conf syntax is ok

nginx: configuration file /opt/nginx/nginx.conf test is successful

[root@localhost opt]# /etc/init.d/nginx reload

Reloading nginx configuration (via systemctl): [ OK ]

TEST LUA:

curl "http://localhost/sum/?a=10&b=20"

Kết quả ra : 30, vậy là Lua đã hoạt động - OK :heart\_eyes:

**Bước 5 : Cấu hình log Header/Body**

Cấu hình full nginx.conf

1. #user  nobody;
2. worker\_processes  auto;
4. events {
5. worker\_connections  1024;
6. }
8. http {
9. include       mime.types;
10. default\_type  application/octet-stream;
12. log\_format log\_req\_resp '$remote\_addr - $remote\_user [$time\_local] '
13. ' "$request" $status $body\_bytes\_sent ${request\_time}ms '
14. '| PRINT\_REQUEST\_BODY: $request\_body '
15. '| PRINT\_REQUEST\_HEADER:"$req\_header" '
16. '| PRINT\_RESPONSE\_HEADER:"$resp\_header" '
17. '| PRINT\_RESPONSE\_BODY:"$resp\_body" ';
19. access\_log  logs/access.log log\_req\_resp;
21. sendfile        on;
22. keepalive\_timeout  65;
24. **#Cấu hình web\_server listen Port\_80**
25. server {
26. listen       80;
27. server\_name  \_;
29. #Step1: Test Lua Sum
30. location /sum {
31. content\_by\_lua\_block {
32. local args = ngx.req.get\_uri\_args();
33. ngx.say(args.a + args.b)
34. }
35. }
36. location / {
37. #Step2: Config REPSONSE\_BODY
38. lua\_need\_request\_body on;
40. set $resp\_body "";
41. body\_filter\_by\_lua '
42. local resp\_body = string.sub(ngx.arg[1], 1, 1000)
43. ngx.ctx.buffered = (ngx.ctx.buffered or "") .. resp\_body
44. **if** ngx.arg[2] then
45. ngx.var.resp\_body = ngx.ctx.buffered
46. end
47. ';
49. #Step3: Config REQUEST\_HEADER, RESPONSE\_HEADER
50. set $req\_header "";
51. set $resp\_header "";
52. header\_filter\_by\_lua '
53. local h = ngx.req.get\_headers()
54. **for** k, v in pairs(h) **do**
55. ngx.var.req\_header = ngx.var.req\_header .. k.."="..v.." "
56. end
57. local rh = ngx.resp.get\_headers()
58. **for** k, v in pairs(rh) **do**
59. ngx.var.resp\_header = ngx.var.resp\_header .. k.."="..v.." "
60. end
61. ';
63. proxy\_pass http://localhost:8080/;    # ở đây tôi proxy vào 1 con tomcat8 test
64. }


68. error\_page   500 502 503 504  /50x.html;
69. location = /50x.html {
70. root   html;
71. }
72. }
73. }

Kiểm tra test lại nginx #nginx\_syntax

[root@localhost nginx]# nginx -t

nginx: the configuration file /opt/nginx/nginx.conf syntax is ok

nginx: configuration file /opt/nginx/nginx.conf test is successful

[root@localhost opt]# /etc/init.d/nginx reload

Reloading nginx configuration (via systemctl): [ OK ]

Bước: Test POST chả hạn

[root@localhost ROOT]# curl -X POST -H "Host:tuanduong122.wordpress.com" -H "Tuanda:hehehe" -d '{"DUONGANHTUAN\_BODY":"123"}' localhost

Kết quả: APACHE TOMCAT8 BODY

vào đọc log accesss:

tail /opt/nginx/logs/access.log

127.0.0.1 - - [24/Apr/2020:19:04:26 -0400] "POST / HTTP/1.1" 200 20 0.346ms | PRINT\_REQUEST\_BODY: {\x22DUONGANHTUAN\_BODY\x22:\x22123\x22} | PRINT\_REQUEST\_HEADER:"host=tuanduong122.wordpress.com content-type=application/x-www-form-urlencoded tuanda=hehehe accept=\*/\* content-length=27 user-agent=curl/7.29.0 " | PRINT\_RESPONSE\_HEADER:"content-length=20 set-cookie=JSESSIONID=C94254D1F47723462F60222FE865D9FB; Path=/; HttpOnly content-type=text/html;charset=ISO-8859-1 connection=keep-alive " | PRINT\_RESPONSE\_BODY:"APACHE TOMCAT8 BODY\x0A"

Phân tích Kết quả:

**PRINT\_REQUEST\_BODY:** {\x22DUONGANHTUAN\_BODY\x22:\x22123\x22}

**PRINT\_REQUEST\_HEADER:**"host=tuanduong122.wordpress.com content-type=application/x-www-form-urlencoded tuanda=hehehe accept=*/* content-length=27 user-agent=curl/7.29.0 "

**PRINT\_RESPONSE\_HEADER:**"content-length=20 set-cookie=JSESSIONID=C94254D1F47723462F60222FE865D9FB; Path=/; HttpOnly content-type=text/html;charset=ISO-8859-1 connection=keep-alive "

**PRINT\_RESPONSE\_BODY:**"APACHE TOMCAT8 BODY\x0A"

**Other: Lua Show Request Body as Response**

location = /request\_body {

client\_max\_body\_size 50k;

client\_body\_buffer\_size 50k;

content\_by\_lua\_block {

ngx.req.read\_body() -- explicitly read the req body

local data = ngx.req.get\_body\_data()

if data then

ngx.say("body data:")

ngx.print(data)

return

end

-- body may get buffered in a temp file:

local file = ngx.req.get\_body\_file()

if file then

ngx.say("body is in file ", file)

else

ngx.say("no body found")

end

}

}

[root@tuanda conf]# curl -X GET -d '{test:123}' localhost/request\_body

body data:

{test:123}

Lua\_Nginx : <https://github.com/openresty/lua-nginx-module>