**Nginx+Lua笔记**

**Lua.**

Lua 是一个简洁，轻量,可扩展的脚本语言：

[Nginx 配置支持 WAF](https://www.cnblogs.com/Template/p/9668305.html)

WAF（Web Application Firewall），中文名叫做“Web应用防火墙”

WAF的定义是这样的：Web应用防火墙是通过执行一系列针对HTTP/HTTPS的安全策略来专门为Web应用提供保护的一款产品,通过从上面对WAF的定义中，

我们可以很清晰地了解到：WAF是一种工作在应用层的、通过特定的安全策略来专门为Web应用提供安全防护的产品。

ngx\_lua\_waf是一个基于ngx\_lua的web应用防火墙。

**# ngx\_lua\_waf用途：**

防止sql注入，本地包含，部分溢

出，fuzzing测试，xss，SSRF等web攻击

　　防止svn/备份之类文件泄露

　　防止ApacheBench之类的压力测试工具的攻击

　　屏蔽常见的黑客扫描工具，扫描器

　　屏蔽异常的网络请求

　　屏蔽图片附件类目录PHP执行权限

防止webshell上传

默认情况下Nginx不支持Lua模块，需要安装LuaJIT解释器，并且需要重新编译Nginx，可选择使用openrestry open ruai si ti

Ngx\_devel\_kit： 一个拓展nginx服务器核心功能的模块，第三方模块开发可以基于它来快速实现

lua\_nginx\_module

**LuaJIT 简单了解**

[root@web01 ~]# rpm -qa lua

lua-5.1.4-15.el7.x86\_64

**交互式**

[root@web01 ~]# lua

Lua 5.1.4 Copyright (C) 1994-2008 Lua.org, PUC-Rio

> print ("hello word")

hello word

**非交互式**

[root@web01 ~]# cat test.lua

#!/usr/bin/lua

print ("hello world")

[root@web01 ~]# lua test.lua

hello world

**Lua语言注释以及变量**

**[root@web01 ~]# cat test.lua**

#!/usr/bin/lua

--Date 2019-02-26  #行注释

print ("hello world")

--[[

lin yao hong #块注释

Date 2019-02-26

--]]

a = 123 #变量

print ("a",a)

**[root@web01 ~]# lua test.lua**

hello world

a 123

[root@localhost ~]# hostnamectl set-hostname nginx\_lua

[root@localhost ~]# bash

**安装依赖包：**

yum -y install gcc gcc-c++ autoconf automake make unzip

yum -y install zlib zlib-devel openssl openssl-devel pcre pcre-devel

**下载所用到的依赖包、并且解压**

cd /server/tools/

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

wget <https://github.com/simplresty/ngx_devel_kit/archive/v0.3.0.tar.gz>

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

tar zxvf LuaJIT-2.0.5.tar.gz

tar zxvf v0.10.13.tar.gz

tar zxvf v0.3.0.tar.gz

**安装LuaJIT2.0：****LuaJIT是采用C语言写的Lua代码的解释器**

cd LuaJIT-2.0.5

make && make install

ln -s /usr/local/lib/libluajit-5.1.so.2 /lib64/liblua-5.1.so.2

ls -l /lib64/liblua-5.1.so.2

echo "/usr/local/lib" >> /etc/ld.so.conf

如果不创建符号链接，可能出现以下异常：

error while loading shared libraries: libluajit-5.1.so.2: cannot open shared object file: No such file or directory

导入环境变量（其他方式）：

echo "export LUAJIT\_LIB=/usr/local/lib" >> /etc/profile

echo "export LUAJIT\_INC=/usr/local/include/luajit-2.0" >> /etc/profile
source /etc/profile

**开始安装Nginx**

wget <http://10.10.10.100/src/nginx-1.14.2.tar.gz>

cd nginx-1.14.2

./configure --prefix=/application/nginx-1.14.2 --user=www --group=www --with-http\_stub\_status\_module --with-http\_v2\_module --with-http\_ssl\_module --with-http\_gzip\_static\_module --with-http\_realip\_module --with-http\_flv\_module --with-http\_mp4\_module --with-openssl=../openssl-1.0.2q --with-pcre=../pcre-8.42 --with-pcre-jit --with-ld-opt='-ljemalloc' --with-zlib=../zlib-1.2.11 --with-file-aio --with-http\_dav\_module --with-http\_addition\_module **--add-module=../lua-nginx-module-0.10.13/** **--add-module=../ngx\_devel\_kit-0.3.0/**

make

make install

**至此安装完成、首先测试是否安装成功**

location /waftest {

default\_type text/html;

content\_by\_lua\_block {

ngx.say("HelloWord")

}

}

如下图：代表安装成功



**配置和使用WAF防护功能**

**把ngx\_lua\_waf下载到conf目录下,解压命名为waf**

cd /server/application/nginx/conf/

wget <https://github.com/loveshell/ngx_lua_waf/archive/master.zip>

unzip master.zip

mv ngx\_lua\_waf-master/ waf

**在nginx.conf的http段添加下面这段**

include luawaf.conf;

然后在/server/application/nginx/conf/目录下创建: luawaf.conf;

内容为：

lua\_package\_path "/server/application/nginx/conf/waf/?.lua";

lua\_shared\_dict limit 10m;

init\_by\_lua\_file /server/application/nginx/conf/waf/init.lua;

access\_by\_lua\_file /server/application/nginx/conf/waf/waf.lua;

**或者直接在http段添加如下内容：**

lua\_package\_path "/server/application/nginx/conf/waf/?.lua";

lua\_shared\_dict limit 10m;

init\_by\_lua\_file /server/application/nginx/conf/waf/init.lua;

access\_by\_lua\_file /server/application/nginx/conf/waf.lua;

**修改**

[root@localhost /server/application/nginx/conf/waf]# pwd

/server/application/nginx/conf/waf

[root@localhost /server/application/nginx/conf/waf]# vim config.lua

RulePath = "/server/application/nginx/conf/waf/wafconf/"

attacklog = "on"

logdir = "/server/logs/waf/"

**攻击日志目录，并赋予nginx用户对该目录的写入权限**

cd /server/logs/

chown www:www waf/

访问

[root@work ~]# curl http://10.10.10.26/test.php?id=../etc/passwd

11111

PS：**个性化修改部分：**

[root@localhost /server/application/nginx/conf/waf]# vim init.lua +66

加入：html=read\_rule('html')

把config.lua里的html相关内容删除掉。在wafconf目录里创建html文件（没什么卵用，就是方便）

[root@localhost /server/application/nginx/conf/waf/wafconf]# ll -l html

-rw-r--r-- 1 root root 468 2019-02-26 17:29 html



1、漏洞扫描器

AppScan

WebInspect

NetSparker

Websecurify

WebCruiser

Nikto

wikto

w3af

vega

OWASP-ZAP

arachni-scanner

golismero

brakeman ruby on rails漏洞扫描器

grendel-scan

(2)专项扫描器

a. SQLI/NoSQLI

Havij, SQLMap, Pangolin

b.XSS

X5S，XSScrapy

c.文件包含

fimap

d.开源框架漏洞扫描器

wpscan

joomscan

(3)密码破解工具

hydra

medusa

patator

(4)目录字典攻击工具

Pker

dirbuster

(5)其他

burpsuit

MSF