链滴

云主机安全——nginx_lua_waf

作者：yuanhenglizhen

原文链接：https://ld246.com/article/1550857969884

来源网站：链滴

沐风

这边需要注意的是路径写错了会报lua的错误 nginx 500

1.安装openresty

docker run -d -p 80:80 -p 443:443 --restart=always -v /data/prod/openresty/conf.d:/etc/ngix/conf.d -v /data/prod/openresty/html:/usr/share/nginx/html -v /data/prod/openresty/logs:/ar/log/nginx -v /data/prod/openresty/nginx.conf:/usr/local/openresty/nginx/conf/nginx.conf:o --name=nginx openresty/openresty:centos

2.https证书

acme.sh --issue --dns dns_ali -d .xuuo.com --installcert --key-file /data/prod/openresty/conf.d/.uuo.com.key --fullchain-file /data/prod/openresty/conf.d/*.xuuo.com.pem--reloadcmd "docker restart nginx"

/root/.acme.sh/acme.sh --issue -d *.xuuo.com --dns dns_ali --force

3.nginx加上wtf安全过滤

此处的路径以实际路径为准，容器部署的以容器具体路径为准

1.安装并配置WAF：

#git clone https://github.com/unixhot/waf.git

#cp -a ./waf/waf /usr/local/openresty/nginx/conf/

修改Nginx的配置文件，在HTTP字段加入以下配置。注意路径，同时WAF日志默认存放在/tmp/日期waf.log

vim /usr/local/openresty/nginx/conf/nginx.conf

#WAF

lua_shared_dict limit 50m;

lua_package_path "/usr/local/openresty/nginx/conf/waf/?.lua";

init_by_lua_file "/usr/local/openresty/nginx/conf/waf/init.lua";

access_by_lua_file "/usr/local/openresty/nginx/conf/waf/access.lua";

[root@openstack-compute-node5 ~]# /usr/local/openresty/nginx/sbin/nginx –t

[root@openstack-compute-node5 ~]# /usr/local/openresty/nginx/sbin/nginx

2.WAF配置文件：

vim /usr/local/openresty/nginx/conf/waf/config.lua

--waf 是否开启

config_waf_enable = "on"

--日杂文件目录

config_log_dir = "/tmp"

--配置文件目录

config_rule_dir = "/usr/local/openresty/nginx/conf/waf/rule-config"

--是否开启 白名单链接

config_white_url_check = "on"

--enable/disable 白名单IP

config_white_ip_check = "on"

--enable/disable 黑名单IP

config_black_ip_check = "on"

--enable/disable URL检测

config_url_check = "on"

--enalbe/disable url 参数检查

config_url_args_check = "on"

--enable/disable user agent filtering

config_user_agent_check = "on"

--enable/disable cookie deny filtering

config_cookie_check = "on"

--enable/disable cc 检测

config_cc_check = "on"

--CC检测限制60秒内同一URL只能访问10次

config_cc_rate = "10/60"

--enable/disable post 检测（这个功能作者没完成）

config_post_check = "on"

--config waf output redirect/html

config_waf_output = "html"

3.验证：

http://21.21.34.5/select * from #访问会出现安全检测页面

ab -n100 -c1 http://172.16.1.211/ #模仿CC攻击

测试

成功拦截