1. nodeJS服务器封一切端口仅仅保留nginx server IP
2. nginx server限制并发数

http {

limit\_conn\_zone $binary\_remote\_addr zone=addr:10m;

server {

listen 80;

server\_name www.tomener.com tomener.com;

location / {

root /var/www/tomener;

index index.php index.html index.htm;

limit\_conn addr 5; #是限制每个IP只能发起5个连接

limit\_rate 100k; #限速为 100KB/秒

}

}

1. nginx server封IP

location/ {   
   deny  192.168.1.1;        封192.168.1.1  
   deny 192.168.1.0/24;   封192.168.1.0-192.168.1.255 IP段  
   deny  all;   封所有IP  
}

要仅允许你某特定IP，

Allow 192.168.1.1 注意在vmware虚拟机中应当封虚拟网卡中的地址

Deny all

1. 其它模块：

ModSecurity 应用层WAF，功能强大，能防御的攻击多，配置复杂

ngx\_lua\_waf 基于ngx\_lua的web应用防火墙，使用简单，高性能和轻量级

http\_guard 基于openresty

1. 隐藏不必要的信息

server\_tokens off; 隐藏版本号防止特定版本bug

proxy\_hide\_header X-Powered-By; 隐藏使用nginx服务器

1. 禁止非必要的方法

if ($request\_method !~ ^(GET|HEAD|POST)$ ) {

return 444;

} //禁止TRACE方法

1. 配置http响应头

add\_header Strict-Transport-Security "max-age=31536000";

add\_header X-Frame-Options deny;

add\_header X-Content-Type-Options nosniff;

Strict-Transport-Security（简称为 HSTS）可以告诉浏览器，在指定的 max-age 内，始终通过 HTTPS 访问。即使用户自己输入 HTTP 的地址，或者点击了 HTTP 链接，浏览器也会在本地替换为 HTTPS 再发送请求

X-Frame-Options 用来指定此网页是否允许被 iframe 嵌套，deny 就是不允许任何嵌套发生。X-Content-Type-Options 用来指定浏览器对未指定或错误指定 Content-Type 资源真正类型的猜测行为，nosniff 表示不允许任何猜测。

Content-Security-Policy（简称为 CSP）用来指定页面可以加载哪些资源，主要目的是减少 XSS 的发生。我允许了来自本站、disquscdn 的外链 JS，还允许内联 JS，以及在 JS 中使用 eval；允许来自本站和 google 统计的图片，以及内联图片（Data URI 形式）；允许本站外链 CSS 以及内联 CSS；允许 iframe 加载来自 disqus 的页面。对于其他未指定的资源，都会走默认规则 self，也就是只允许加载本站的。

1. 启用https安全配置

启用 HTTPS 并正确配置了证书，意味着数据传输过程中无法被第三者解密或修改。有了 HTTPS，也得合理配置好 Web Server，才能发挥最大价值。

1）使用semanage管理、查看、注册port端口

安装semanage: yum install policycoreutils-python

查看当前哪些受控端口可执行：semanage port -l | grep '^http\_port\_t'

2）安装openssl工具

yum install openssl

yum install openssl-devel

3）将ssl证书颁发给自己

mkdir /etc/ssl/private

openssl genrsa -des3 -out server.key 1024

openssl req -new -key server.key -out server.csr

openssl rsa -in server.key -out server\_nopwd.key

openssl x509 -req -days 365 -in server.csr -signkey server\_nopwd.key -out server.crt

一路OK即可

4）配置nginx https server

编辑/etc/nginx/nginx.conf文件

server {

listen 443 ssl;

listen 8080 default;

ssl\_certificate /etc/ssl/private/server.crt;

ssl\_certificate\_key /etc/ssl/private/server\_nopwd.key;

}

可访问

<http://IP:8080>

<https://IP:443>

https://IP

由于nginx默认使用no password的key,所以在nginx.conf中使用server\_nopwd.key

5）https默认使用443端口，确认此端口已注册。

semanage port -l | grep 443

6）确认443端口已被监听

netstat -anp | grep 443

或查看所有监听端口 netstat –anltp

7）如果网站URL是：<http://ip:port/>，如果再在网站上启用ssl，在浏览器上访问<https://ip:port/>，将会收到如下错误： ssl\_error\_rx\_record\_too\_long

原因：ssl协议默认是在web server的443端口监听，所以访问ssl会产生如下形式的访问：<http://ip:443/>，这样将与<http://ip:port/>里的port冲突。

对https访问，web server会自己定位到导用的了ssl的网站上。

解决方法：

对<https://ip:port/>的访问可以省略port，如<https://ip/>

1. 只允许https访问，禁止http访问