# 基于 nginx 和 modsecurity 的 WAF 防火墙实现

## 第一部分 简单规则

引擎：使用 nginx 内置变量及正则表达式实现
作用范围：站点 server 字段
有部分规则重复进行注释处理，当 modsecurity 模块不可用时取消注释使其生效
1.过滤文件和路径
阻止 /~ 这种带有波浪线的路径
#阻止文件类型（扩展名、后缀）
.(bzr|cvs|git|svn)
.(bak|backup|bzr|cfg|conf|cvs|doc|docx|DS_Store|ear|git|gitignore|hg|htaccess|htpasswd|ini|inc|jar|log|online|production|project|properties|pl|pm|py|pyc|pyo|sh|sql|svn|swp|war)$
#阻止常见 windows 文件格式
.(ade|adp|app|asa|ascx|ashx|asmx|asp|aspx|axd|bas|bat|cdx|cer|chm|class|cmd|com|config|cpl|crt|cs|csproj|csh|csr|dat|dbf|dll|dos|exe|fxp|hlp|hta|htr|htw|ida|idc|idq|ins|isp|its|jse|key|ksh|licx|lnk|mad|maf|mag|mam|maq|mar|mas|mat|mau|mav|maw|mda|mdb|mde|mdt|mdw|mdz|msc|msh|msh1|msh1xml|msh2|msh2xml|mshxml|msi|msp|mst|old|ops|pass|pcd|pdb|pif|pol|prf|prg|printer|pst|pwd|resources|resx|reg|rem|scf|scr|sct|shb|shs|shtm|shtml|soap|stm|sys|url|vb|vbe|vbs|vbproj|vsdisco|webinfo|xsd|xsx|ws|wsc|wsf|wsh)$
2.过滤 http 请求方法 仅允许 GET HEAD POST OPTIONS
3.过滤用户代理
阻止各种机器人（robot），爬虫（spider），下载器，测试工具，注入工具，扫描器
BTWebClient|FlashGet|FreshDownload|JetCar|PycURL|wget
audit|BabyKrokodil|BBBike|httrack|httperf|harvest|hydra|netsparker|Nikto|owasp|parser
Alligator|Azureus|BackStreet Browser|BW-C-2.0|Charon|LWP::Simple
ApacheBench|GetRight|github|GrabNet|Havij|Jmeter|JoeDog|masscan|mail2000|TurnitinBot|WebBench
CPython|libwww|libwww-perl|python-httplib2|python-requests|Python-urllib
arachni|absinthe|bilbo|black widow|blackwidow|brutus|bsqlbf|cgichk|dirbuster|fimap|grabber|grendel-scan|havij|hydra|jaascois|jbrofuzz|libwhisker|metis|n-stealth|netsparker|nasl|nmap|nse|nsauditor|nikto|nessus|Openvas|pmafind|paros|pangolin|sqlmap|sqlninja|sql power injector|webinspect|wifinder|w3af|whatweb|webtrends security analyzer|webshag|WinHttp
AhrefsBot|AltaVista|aiHitBot|BBScan|BLEXBot|CSS Certificate Spider|COMODO SSL Checker|Dataprovider|
electricmonk|eMusic|Exabot|FeedBurner|Feedskycrawler|ia_archiver|ips-agent|NgSpider|panscient.com|Plukkie|SemrushBot|Seznam Bot|spiderman|seoscanners.net|SafeDNSBot|scrapbot|SurveyBot|semanticbot|SiteExplorer|Scrapy|Uptimebot|Wotbox|YRSpider

4.过滤变量:强制规范特定类型变量，比如禁止数值变量传递文本字串

~~5.过滤 SQL 注入~~
~~6.过滤 XSS 跨域~~
7.过滤 referer
只允许 http://和 https://开头的 referer
8.防止快速 DOS 攻击规则
全局并发请求限制不区分内容，特定内容由 modsecurity 模块处理
~~每个客户端 IP 100 并发~~
~~每个服务器域名 2000 并发~~
全局请求速率限制不区分内容，特定内容由 modsecurity 模块处理
~~每个客户端 IP 600 次/分钟~~
~~每个服务器域名 2000 次/秒~~
全局限速规则
~~前 100MB 不限速，超过 100MB 后限速 4KB/s~~
注： Pf(packet filter)防火墙： 系统底层限速，优先级，抗 DOS 攻击,内核代码完成同类工作比 nginx 更健壮更高效

## 第二部分 高级规则

引擎：由第三方模块 modsecurity 实现
作用范围：location 字段静态页面和动态页面
已明确定义的静态资源不做过滤
web 服务器指纹伪装，迷惑入侵者
SecServerSignature `"Apache/2.4.25 (HardenedBSD) PHP/7.1.1"`

# Modsecurity 规则集详单

## 基础规则(base_rules)

HTTP 协议规范(20_protocol_violations)
Id: 960911
符合 HTTP RFC 规范的完整 URL 请求格式
"http:" "//" host [ ":" port ] [ abs_path [ "?" query ]]
"https:" "//" host [ ":" port ] [ abs_path [ "?" query ]]
http://www.w3.org/Protocols/rfc2616/rfc2616-sec3.html#sec3.2.1
http://capec.mitre.org/data/definitions/272.html
id: 981227
识别被 Apache 阻止的无效 URI
Id: 960000
识别 multipart/form-data name 名称绕过企图
检查文件或者文件名变量的元字符(' " ; =)
# https://www.owasp.org/index.php/ModSecurity_CRS_RuleID-960000
# http://www.ietf.org/rfc/rfc2183.txt
Id: 960912
验证请求体已经正确处理

检查 REQBODY_ERROR 变量是否存在
Id: 960914
严格检查 Multipart 解析
id:960915
Multipart 边界不匹配检查
id:960016
内容长度(content length)仅接受数字
http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.13
id:960011
不接受带有 body 的 GET 和 HEAD 请求
http://www.w3.org/Protocols/rfc2616/rfc2616-sec4.html#sec4.3
id:960012
每个 POST 请求必须提供内容长度（Content-Length）
http://www.w3.org/Protocols/HTTP/1.0/spec.html#POST
http://www.w3.org/Protocols/HTTP/1.0/spec.html#Content-Length
id:960902
拒绝压缩内容入站 Identity 只能用于 Accept-Encoding 头，不能用于 Content-Encoding 头
禁止设置：Content-Encoding: Identity
http://www.w3.org/Protocols/rfc2616/rfc2616-sec3.html
id:960022
Expect 头是 http1.1 的功能，自动化程序和机器人通常不遵守 HTTP RFC
Expect 如果含有 100-continue 将被阻止
http://www.bad-behavior.ioerror.us/documentation/how-it-works/
id:960020
Pragma 头里面必须含有 Cache-Control 头
自动化程序和机器人通常不遵守 HTTP RFC
http://www.bad-behavior.ioerror.us/documentation/how-it-works/
id: 958291 958230 958231
Range 头检查：
1. Range 以 0 开头，正规浏览器不会这样做，自动化程序和机器人通常不遵守 HTTP RFC
2. last-byte-pos 必须大于等于 first-byte-pos
3.识别一个请求里多个区间
Id:958295
损坏或者恶意客户端通常有重复或者冲突头，自动化程序和机器人通常不遵守 HTTP RFC
检查 Connection 头多个或者冲突数据
Id: 950107 950109 950108
检查 URL 编码
REQUEST_URI 2 位%0-f　4 位%u0-f
ARGS Content-Type 头 REQUEST_BODY:application/x-www-form-urlencoding XML
Id:950801
检查 UTF 编码，仅用于 UTF-8 编码网站，否则导致失败
Id: 950116
禁止使用全宽 UNICODE，会导致解码绕过
http://www.kb.cert.org/vuls/id/739224

id：960014
代理访问企图
解析 URI 是否指定完整域名并且是否与 SERVER_NAME 匹配，如果不匹配认为客户端请求站外位置
Id:960901 960018
限制发送字符类型：禁止使用 NULL
http://i-technica.com/whitestuff/asciichart.html
ARGS|ARGS_NAMES|REQUEST_HEADERS|!REQUEST_HEADERS:Referer "@validateByteRange 1-255"

HTTP 协议异常(21_protocol_anomalies)
正规浏览器都有 Host, User-Agent 和 Accept 头，如果不全就意味着是攻击者或者自动化客户端
id:960008 960007
Host 头 没有或者空 阻止
id:960015 960021
Accept 头 没有或者空 阻止
id:960009 960006
User-Agent 头 没有或者空 阻止
id:960904
带有 Request Body 却没有 Content-Type 头 阻止
Content-Type 头和 Content-Length 头不能空或者 0
Id:960017
Host 头禁止 IP 地址
http://technet.microsoft.com/en-us/magazine/2005.01.hackerbasher.aspx
id: 960013
HTTP/1.1 POST 请求如果没有 Transfer-Encoding 必须提供 Content-Length
http://httpwg.github.io/specs/rfc7230.html#header.content-length

请求限制(23_request_limits)
大多数情况，需要确定请求的最大容量，比如一个请求 400 个参数
id:960209
参数名长度限制 100 字符
Id:960208
参数值长度限制 400 字符
id:960335
参数数量限制 255 个
Id:960341
参数总长度限制 64000 字符
~~Id:960342~~
~~单个文件大小限制~~
~~Id:960343~~
~~合并文件大小限制~~

HTTP 策略(30_http_policy)

~~id:960032~~

~~请求方法限制 仅允许 GET HEAD POST OPTIONS~~

Id:960010

限制 content-types 仅允许 application/x-www-form-urlencoded|multipart/form-data|text/xml|application/xml|application/x-amf|application/json

Id:960034

限制 HTTP 协议版本 仅允许 HTTP/1.0 HTTP/1.1 HTTP/2.0

~~id:960035~~

~~限制文件扩展名 被禁止扩展名~~

id:960038

限制 HTTP 头

禁止使用 Proxy-Connection Lock-Token Content-Range Translate via if

恶意机器人(35_bad_robots)

id:990002

根据 User-Agent 识别的扫描器 modsecurity_35_scanners.data

Id:990901

REQUEST_HEADERS_NAMES "\bacunetix-product\b"

Id: 990902

REQUEST_FILENAME "@pm nessustest appscan_fingerprint"

id:990012

根据 User-Agent 识别的机器人 modsecurity_35_bad_robots.data

robot 评分列表

常见攻击(40_generic_attacks)

id:950907

系统命令注入攻击(OS Command Injection Attacks)

# http://projects.webappsec.org/OS-Commanding

# http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

阻止命令 cc curl wget

id:960024

识别非文字重复字符(大于等于 4 次)

id:950008

ColdFusion 注入(Coldfusion Injection)

http://www.adobe.com/devnet/security/security_zone/asb99-10.html

阻止无文档的 ColdFusion 标签

id:950010

LDAP Injection

http://technet.microsoft.com/en-us/library/aa996205%28EXCHG.65%29.aspx

阻止 LDAP 注入攻击

id:950011

阻止服务器端包含注入(Server-Site Include)

SSI injection
http://projects.webappsec.org/SSI-Injection
id:950018
通用 PDF 跨域脚本(UPDF XSS)
Universal PDF XSS URL
http://www.modsecurity.org/projects/modsecurity/apache/feature_universal_pdf_xss.html
id:950019
邮件注入(Email Injection)
http://projects.webappsec.org/Mail-Command-Injection
id:950012
HTTP Request Smuggling
http://projects.webappsec.org/HTTP-Request-Smuggling
# http://article.gmane.org/gmane.comp.apache.mod-security.user/3299

id:950910 950911
HTTP Response Splitting
http://projects.webappsec.org/HTTP-Response-Splitting
id:950117 950118 950119 950120
RFI Attack
#
# -=[ Rule Logic ]=-
# These rules look for common types of Remote File Inclusion (RFI) attack methods.
#  - URL Contains an IP Address
#  - The PHP "include()" Function
#  - RFI Data Ends with Question Mark(s) (?)
#  - RFI Host Doesn't Match Local Host
#
# -=[ References ]=-
# http://projects.webappsec.org/Remote-File-Inclusion
#  http://tacticalwebappsec.blogspot.com/2009/06/generic-remote-file-inclusion-attack.html
id:950121
限制发送的字符类型:禁止 NULL
id:981133 981134
净化请求方法
id:950009 950003 950000
会话固定攻击
Begin RegEx Checks for target locations that matched the prequalifier checks
# Session fixation
http://projects.webappsec.org/Session-Fixation

id:950005
文件注入(File Injection)

id:950002
命令访问(Command access)
id:950006
命令注入(Command injection)
id:959151 958976 958977
PHP 注入(PHP injection)

SQL 注入攻击 41_sql_injection_attacks
SQL Injection Pocket Reference (via @LightOS) -
# https://docs.google.com/Doc?docid=0AZNIBave77hiZGNjanptbV84Z25yaHJmMjk
SQLi Filter Evasion Cheat Sheet -
# http://websec.wordpress.com/2010/12/04/sqli-filter-evasion-cheat-sheet-mysql/
SQL Injection Cheat Sheet -
# http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
SQLMap's Tamper Scripts (for evasions)
# https://svn.sqlmap.org/sqlmap/trunk/sqlmap/tamper/
id:981231
探测 SQL 注释序列
id:981260
SQL Hex Evasion Methods
id:981318
String Termination/Statement Ending Injection Testing
id:981319
SQL 运算符
id:950901
SQL Tautologies
id:981320
探测数据库名称
Id: 981300-981317
SQL 关键词异常评分
id:950007
SQL 盲注（Blind SQL injection）
id:950001 959070 959071 959072 950908 959073
SQL 注入（SQL injection）
Id:981172 981173
SQL 注入字符异常使用
id:981272 981244 981255 981257 981248 981277 981250 981241 981252 981250 981241 981252 981256 981245 981276 981254 981270 981240 981249 981253 981242 981246 981251 981247 981243
PHPIDS - Converted SQLI Filters
https://dev.itratos.de/projects/php-ids/repository/raw/trunk/lib/IDS/default_filter.xml

XSS（跨域）攻击(41_xss_attacks)
id:973336

XSS Filters - Category 1
script tag based XSS vectors
id:973337
XSS Filters - Category 2
XSS vectors making use of event handlers like onerror, onload etc
id:973338
XSS Filters - Category 3
XSS vectors making use of Javascripts URIs
id:981018 958414 958032 958026 958027 958054 958418 958034 958019 958013 958408 958012 958423 958002 958017 958007 958047 958410 958415 958022 958405 958419 958028 958057 958031 958006 958033 958038 958409 ···
XSS

id:973300 973301
Detect tags that are the most common direct HTML injection points
id:973303
Detect event handler names
id:973304
Detect usage of common URI attributes
……

严密安全(42_tight_security)
id:950103
目录遍历攻击 (Directory Traversal)
~~id:950103~~
~~Weaker signature~~

木马(webshell) (45_trojans)
id:950921
id:950922

常规异常(47_common_exceptions)
id:981020
Exception for Apache SSL pinger
id:981021
Exception for Apache internal dummy connection
id:981022
Exception for Adobe Flash Player
https://www.modsecurity.org/tracker/browse/CORERULES-57

~~本地异常(48_local_exceptions)~~

进站阻断(49_inbound_blocking)
id:981175

Alert and Block based on Anomaly Score and OSVDB Check
id:981176
Alert and Block based on Anomaly Score

出站检查(50_outbound)
发生信息泄露阻断访问
id:970007
Zope 信息泄露
id:970008
ColdFusion 信息泄露
id:970009
PHP 信息泄露
id:970010
ISA 服务器存在被公开
id:970012
微软 Office 文档属性泄露
id:970903
ColdFusion 源代码泄露
id:970018
IIS 默认位置
id:970901 970118
应用不可用
阻止报告 5xx 代码
id:970021
Weblogic 信息公开
id:970011
文件或目录名泄露
id:981000 981001
IFrame 注入(IFrame Injection)
id:981004 981005 981006 981007
Generic Malicious JS Detection
id:981178
Run PM check against response body data before running any RegEx Checks
id:970014
ASP/JSP 源代码泄露
id:970015 970902
PHP 源代码泄露
id:970002
统计页面泄露
id:970003
SQL 错误泄露
id:970004 970904
IIS 错误泄露
id:970013

目录列表

出站阻断 59_outbound_blocking
id:981200
异常评分过高发出警告或者阻断，这将阻止出站信息泄露

相关性 60_correlation
Correlated Successful Attack
id:981202
Correlated Attack Attempt

## 试验规则(experimental_rules)

~~暴力破解(11_brute_force)~~

~~id:981036~~

~~Enforce an existing IP address block and log only 1 time/minute~~

~~id:981037~~

~~Block and track # of requests but don't log~~

~~id:981038 981039 981040~~

~~skipAfter Checks~~

~~id:981041~~

~~Brute Force Counter~~

~~id:981042~~

~~Check Brute Force Counter~~

~~id:981043~~

~~Check Brute Force Burst Counter and set Block~~

DOS(拒绝服务)保护(11_dos_protection)
防护策略：某个 IP 每分钟访问(http request)受保护的页面超过 100 次，被屏蔽 10 分钟，超时后自动恢复。常见静态文件和样式不受保护。
id:981044
Anti-Automation rule set for detecting Denial of Service Attacks
id:981045
Block and track # of requests but don't log
id:981046
skipAfter Check
id:981047
DOS Counter
id:981048
Check DOS Counter
id:981049
Check DOS Burst Counter and set Block

代理滥用(11_proxy_abuse)

id:981050

禁止跨国使用代理

Rule set for detecting Open Proxy Abuse/Chaining.

http://blog.spiderlabs.com/2011/03/detecting-malice-with-modsecurity-open-proxy-abuse.html

~~慢速 DOS 保护(11_slow_dos_protection)~~

~~Rule set for detecting Slow HTTP Denial of Service Attacks~~

~~http://blog.spiderlabs.com/2010/11/advanced_topic_of_the_week_mitigating_slow_http_dos_attacks.html~~

~~id:981051~~

~~id:981052~~

~~扫描器集成 16_scanner_integration~~

~~id:900030~~

~~Disable ModSecurity For Arachni Scans~~

~~信用卡追踪和 PAN 泄露检查 25_cc_track_pan~~

~~id:920021 920022 920023~~

~~Credit Card Track 1 and 2 and PAN Leakage Checks~~

~~Credit Card Track 2 Data Leakage~~

~~Credit Card PAN Data Leakage~~

~~应用传感器探测点_设置(40_appsensor_detection_point_2.0_setup)~~

~~id:981082~~

~~应用传感器探测点_请求异常(40_appsensor_detection_point_2.1_request_exception)~~

~~应用传感器探测点_蜜罐陷阱(40_appsensor_detection_point_2.9_honeytrap)~~

~~id:981131~~

~~HT1: Alteration to Honey Trap Data~~

~~https://www.owasp.org/index.php/AppSensor_DetectionPoints#HT1~~

~~id:981132~~

~~Add a fake "debug" hidden parameter to forms~~

~~应用传感器探测点_结尾(40_appsensor_detection_point_3.0_end)~~

HTTP 参数污染(40_http_parameter_pollution)

id:900032

http://tacticalwebappsec.blogspot.com/2009/05/http-parameter-pollution.html

~~CSP 强制执行 42_csp_enforcement~~

~~Content Security Policy (CSP) Settings~~

~~id:981142~~

~~https://developer.mozilla.org/en/Security/CSP~~

~~id:'960001~~

Check the REQUEST_BODY for CSP Violation Report data and generate an Alert

id:'960002

id:'960003

扫描器集成 46_scanner_integration

id:'999003

id:'999004

贝叶斯分析 48_bayes_analysis

id:'900033

id:'900034

id:'900035

响应分析(55_response_profiling)

id:'981188

niframes|nscripts|nlinks|nimages

PVI 检查 56_pvi_checks

Osvdb -

Passive Vulnerabilty Check with OSVDB

IP 取证(61_ip_forensics)

id:900039

Check the Transactional Anomaly Score - if it is not 0 then record the GeoIP data

# for the client in the audit log.

## 可选规则(optional_rules)

忽略静态文件(10_ignore_static)

Determine actions based on static file extensions

图片 jpeg png gif ico、文档 doc pdf txt xls 、HTMLcss js html、媒体文件 mp3 avi flv swf wma

AVS 流量 11_avs_traffic

This ruleset allows you to control how ModSecurity will handle traffic originating

# from Authorized Vulnerability Scanning (AVS) sources.

# See related blog post -

# http://blog.spiderlabs.com/2010/12/advanced-topic-of-the-week-handling-authorized-scanning-traffic.html

启用 XML Body 处理 (13_xml_enabler)

id:981053

The rules in this file will trigger the XML parser upon an XML request

认证追踪 16_authentication_tracking

Create an audit log of a successful Authentication

Create an alert when a user fails authenticating

会话劫持(16_session_hijacking)

~~This rule file will identify outbound Set-Cookie/Set-Cookie2 response headers and~~
~~# then initiate the proper ModSecurity session persistent collection (setsid).~~
~~# The rules in this file are required if you plan to run other checks such as~~
~~# Session Hijacking, Missing HTTPOnly flag, etc...~~
~~#~~

~~#~~
~~# This rule set will identify subsequent SessionIDs being submitted by clients in~~
~~# Request Headers.  First we check that the SessionID submitted is a valid one~~

~~This rule will identify the outbound Set-Cookie SessionID data and capture it in a setsid~~

~~用户名追踪 16_username_tracking~~

~~Template rules for login/audit rules.~~

~~Identify/Set the UserID name and collection~~

~~Password Complexity Check~~

~~已知信用卡 25_cc_known~~

~~Detect CC# in input, log transaction and sanitize~~

~~CSA SmartPay~~

~~MasterCard~~

~~Visa~~

## 垃圾评论(42_comment_spam)

Comment spam is an attack against blogs, guestbooks, wikis and other types of
#  interactive web sites that accept and display hyperlinks submitted by
#  visitors. The spammers automatically post specially crafted random comments
#  which include links that point to the spammer's web site. The links
#  artificially increas the site's search engine ranking and may make the site
#  more noticable in search results.

id:981137 981138 981140

使用全球反垃圾实时黑名单过滤 IP（sbl-xbl.spamhaus.org）

Id:958297

使用文件定义垃圾黑名单过滤 user-agent（modsecurity_42_comment_spam.data）

使用正则表达式阻止垃圾 user-agent

Id:999010

参数和参数名含有<http>  或  <https>被阻止

id:950923

Look for 2 ways of posting a link

Id:950020

Look for too many links in an argument (Prone to FPs)

~~CSRF 保护 43_csrf_protection~~

~~CSRF Protections~~

~~杀毒扫描 46_av_scanning~~

~~调用杀毒软件（或脚本/工具）~~

~~跳过出站检查 47_skip_outbound_checks~~

~~Skip outbound inspection on requests for text content which have no parameters~~

~~头部标记 49_header_tagging~~

~~This file will add Request Header Tagging which allows ModSecurity to communicate~~

~~# any event/rule matches it finds with the downstream application server.  The concept~~

~~# is similar to that of Anti-SPAM apps for Email (such as SpamAssassin).~~

~~添加请求头给后续应用服务器~~

~~55_application_defects~~

~~Charset Checks~~

~~http://websecuritytool.codeplex.com/wikipage?title=Checks#charset~~

~~Charset not set~~

~~http://code.google.com/p/browsersec/wiki/Part2#Content_handling_mechanisms~~

~~Charset not explicitly set to UTF-8 in HTML/XML content ]~~

~~#~~

~~#  http://websecuritytool.codeplex.com/wikipage?title=Checks#charset_not_utf8~~

~~# - http://code.google.com/p/browsersec/wiki/Part2#Character_set_handling_and_detection~~

~~Detect charset mismatches between HTTP header and HTML/XML bodies ]~~

~~#~~

~~# - http://websecuritytool.codeplex.com/wikipage?title=Checks#charset-mismatch~~

~~#  http://code.google.com/p/browsersec/wiki/Part2#Character_set_handling_and_detection~~

~~Cookie Checks ]=-~~

~~#~~

~~# - http://websecuritytool.codeplex.com/wikipage?title=Checks#cookies~~

~~##############################################################################~~

~~#~~

~~# [ Look for cookies with loosely scoped domain restrictions ]~~

~~#~~

~~#  http://websecuritytool.codeplex.com/wikipage?title=Checks#cookie_loosely_scoped_domain~~

~~# - http://code.google.com/p/browsersec/wiki/Part2#Same-origin_policy_for_cookies~~

~~Cookie's HttpOnly Flag Was Not Set ]~~

~~#~~

~~#  http://websecuritytool.codeplex.com/wikipage?title=Checks#cookie_not_setting_httponly_flag~~

~~# - https://www.owasp.org/index.php/HttpOnly~~

~~Fix Missing "httponly" Flag~~

~~Cookie's Secure Flag Was Not Set~~

~~Fix Missing "secure" Flag~~

~~HTTP Header Checks~~

~~Check that the cache-control HTTP header is set to 'no-store'~~

~~Check that a Content-Type header is included in the HTTP response~~

~~Check that IE's XSS protection filter is not being disabled by the Web application~~

~~Check that the X-FRAME-OPTIONS header is being set for Clickjacking defense~~

~~Checks that the X-CONTENT-TYPE-OPTIONS defense against MIME-sniffing has been declared~~

~~XSS Detection - Missing Output Encoding~~

~~Identifies Reflected XSS~~

~~# If malicious input (with Meta-Characters) is echoed back in the reply non-encoded.~~

~~Check to see if TX XSS Data is already in the GLOBAL list.~~

~~Identifies Stored XSS~~

## 市场营销(55_marketing)

搜索引擎爬虫管理

允许的搜索引擎: 百度搜索

禁止的搜索引擎: bing 搜索 Yahoo 搜索 Google 搜索 腾讯搜搜 网易有道 搜狗 360 搜索 阿里一淘 即刻搜索 神马搜索(一搜) 宜搜科技 华为赛门铁克蜘蛛 …

~~SLR 规则(slr_rules)~~

~~适用于特定程序的规则~~

~~46_slr_et_joomla_attacks~~

~~46_slr_et_lfi_attacks~~

~~46_slr_et_phpbb_attacks~~

~~46_slr_et_rfi_attacks~~

~~46_slr_et_sqli_attacks~~

~~46_slr_et_wordpress_attacks~~

~~46_slr_et_xss_attacks~~

对开发人员要求

业务程序请求特征要更接近浏览器和真实人类访问请求，容易被安全软件和人员区分特征，不要与黑客工具，扫描器，机器人之类混同

1. App 请求、API 接口请求、定时任务或者其他程序调用

   设置特定 User-Agent 模拟浏览器，样例微信 Mozilla/5.0 (Linux; Android 6.0; KNT-AL10 Build/HUAWEIKNT-AL10) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/37.0.0.0 Mobile MQQBrowser/6.8 TBS/036849 Safari/537.36 MicroMessenger/6.3.27.880 NetType/WIFI Language/en

不要使用 curl，googlebot，apachebench 之类 user-agent ，不要用库函数默认 user-agent 例如 Python-urllib，user-agent 不能空

2. 设置访问超时和重试次数，请求失败报告错误不再发起请求，不要无限重试
3. 数据库和上游服务器开销巨大的操作，比如卡充值和查询剩余流量，人类正常请求不可能每秒 100 次，设置合理的时间间隔，过于频繁直接拒绝
4. 请求尽量使用 GET 方法，POST 方法过滤性能开销过大，参数传递尽量使用 GET，不能用 GET 情况使用 POST
5. 根据规则拦截日志反馈，修改代码，最终完全开启规则也不会被误拦截。

参考：

1. https://github.com/SpiderLabs/owasp-modsecurity-crs

2017.04.18