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|pmafin

d|paros|pangolin|sqlmap|sqlninja|sql power injector|webinspect|wifinder|w3af|whatweb|webtrends security analyzer|webshag|Win

Http

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=0AZNlBave77hiZGNjanptbV84Z25yaHJmMjk>

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~~](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~~](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~~](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~~](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~~

~~GSA 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~~](http://websecuritytool.codeplex.com/wikipage?title=Checks#charset)

~~Charset not set~~

[~~http://code.google.com/p/browsersec/wiki/Part2#Content\_handling\_mechanisms~~](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~~](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~~](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

1. 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
2. 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
3. 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
4. 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