Зборник радова Факултета техничких наука, Нови Сад

УДК: 004.9
ДОИ: https://doi.org/10.24867/20BE27Nedeljkovic

ПРИМЕНА MODSECURITY WAF-А У ЗАШТИТИ ВЕБ АПЛИКАЦИЈА

APPLICATION OF MODSECURITY WAF IN WEB APPLICATION PROTECTION

Леона Недељковић, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – *У овом раду описан је ModSecurity WAF, приказан је формат правила на која се ослања и које конфигурационе директиве подржава. Затим је на примеру једне веб апликације приказано како може да се употреби и какав утицај његова примена има на перформансе апликације.*

Кључне речи: *ModSecurity, Web Application Firewall, веб апликације*

Abstract – *This paper describes the ModSecurity WAF, shows the format of the rules it relies on and the configuration directives it supports. Then, on the example of a web application, it is shown how it can be used and what impact its application has on the performance of the application.*

Keywords: *ModSecurity, Web Application Firewall, Web applications*

1. УВОД

Ослањање на веб апликације је од суштинског значаја у нашим свакодневним активностима као што су онлајн банкарство, резервација карата и дељења података са другима. Веб апликације су развијене да опслужују *HTML* странице засноване на имплементацији на страни сервера која се може урадити у различитим програмским језицима. Обично имају и базу података. Апликације, међутим, садрже и рањивости које нападачи могу да искористе за свој профит [1].

Да би спречили експлоатацију ових рањивости, веб администратори постављају *firewall*. Они раде на слоју веб апликације, детаљно прегледају *HTTP* пакет и сваки појединачни део *HTTP* пакета и траже нападе на веб апликације тако што проналазе злонамерне стрингове и проблеме у конфигурацији употребом различитих техника попут *whitelist, blacklist* и слично [2].

Велики број *WAF*-ова већ постоји, а један од њих је и *Apache ModSecurity* који је и један од најучесталијих. У овом раду биће појашњено шта је *ModSecurity*, који је формат правила на која се ослања као и од којих конфигурационих директива се састоји.

У последњем поглављу биће приказана његова примена на примеру *bWAPP* апликације као и демонстрација неких врста напада из *browser*-а.

На крају ће бити описано какав утицај има на перформансе апликације и који су начини да се перформансе побољшају.

2. WAF - WEB APPLICATION FIREWALL

Web Application Firewall помаже у заштити веб апликација филтрирањем и надгледањем *HTTP* саобраћаја између веб апликације и интернета. Постављањем *WAF*-а испред веб апликације, поставља се штит између веб апликације и интернета. Док *proxy* server штити идентитет клијентске машине коришћењем посредника, *WAF* је врста *reversed proxy*-ја, који штити сервер од излагања тако што прво захтеви клијената пролазе кроз *WAF* пре него што стигну до сервера [3].

WAF функционише кроз скуп правила која се често називају политикама (енгл. *policies*). Оне имају за циљ да штите од рањивости у апликацијама филтрирањем злонамерног саобраћаја и јединствене су и прилагођене за сваки сајт осим основних, универзалних безбедносних захтева као што је примена важећег *HTTP* протокола. Политике *WAF*-а се брзо и лако могу да модификовати како би се омогућио бржи одговор на различите векторе напада [4][5].

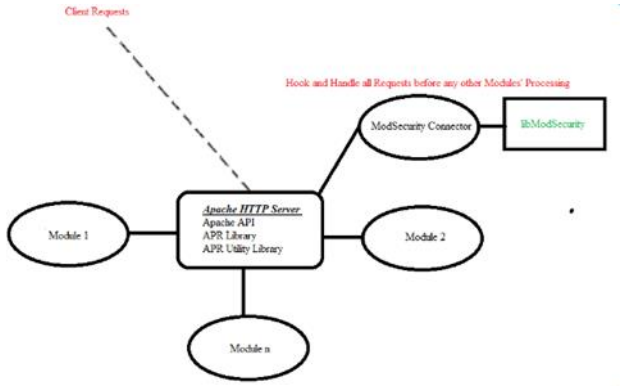
2.1. ModSecurity

ModSecurity је *open-source WAF*. Првобитно дизајниран као модул за *Apache HTTP Server*, еволуирао је да обезбеди низ могућности филтрирања *HTTP* захтева и одговора, заједно са другим безбедносним функцијама на бројним различитим платформама укључујући *Apache HTTP Server, Microsoft IIS* и *Nginx* [6].

ModSecurity модул је написан са крајњим циљем праћења саобраћаја на *Apache HTTP Server*-у. Прва верзија је објављена у новембру 2002. године која је подржавала *Apache HTTP Server 1.3.x*. Будући да је првобитно био *Apache* модул, преношење *ModSecurity*-ја на друге платформе је одузимало много времена и имало високе трошкове одржавања. Као резултат тога, комплетно преписивање је започето у 2015. години. Ова нова итерација, *libmodsecurity*, мења основну архитектуру, одвајајући *ModSecurity* у самостални механизам који комуницира са веб сервером преко *API*-ја. Овај *WAF* заснован на модуларној архитектури, који је најављен за јавну употребу у јануару 2018. године, постао је *libmodsecurity* - *ModSecurity* верзија 3.0 и подржава конекторе за *Nginx* и *Apache*.

НАПОМЕНА:
Овај рад проистекао је из мастер рада чији ментор је био др Горан Сладић, ред. проф.

Стара верзија *ModSecurity*-ја је првобитно била дизајнирана и садржана у *Apache* модулу. Ова тренутна верзија апстрахује неке од детаља омогућавајући *ModSecurity*-ју да лакше подржава више платформи и функција изван обима онога што *Apache* интерни елементи тренутно подржавају. Као резултат тога, коришћење новог *libmodsecurity*-ја може се користити за напајање више различитих конектора што ову верзију чини флексибилнијом [7] [8].
На слици 1 приказана је архитектура *Apache HTTP Server*-а са модулима и конектором за *ModSecurity*. *ModSecurity-Apache* конектор повезује *Apache* и *libmodsecurity*. Другим речима обезбеђује комуникациони канал између њих и он је неопходан да би се могао користити *libmodsecurity* са *Apache*-ом. Овај конектор има облик *Apache* модула [9].



Слика 1. *Архитектура Apache HTTP Server-а са модулима и конектором за ModSecurity [10]*

Функционалности које нуди *ModSecurity* отприлике спадају у четири области [11]:

- **Парсирање**

ModSecurity покушава да искористи што више података који су доступни. Подржани формати података су подржани сигурносно-савесним парсерима који издвајају битове података и чувају их за употребу у правилима.

- **Баферовање**

У типичној инсталацији, баферују се и тела захтева и одговора. То значи да *ModSecurity* обично види комплетне захтеве пре него што се проследе апликацији и комплетне одговоре пре него што се пошаљу клијентима. Баферовање је важна карактеристика, јер је једини начин да се обезбеди поуздано блокирање. Недостатак му је што захтева додатни *RAM* за складиштење података тела захтева и одговора.

- **Логовање**

Потпуна евиденција захтева и одговора је велики део онога што *ModSecurity* ради. Ова функција нам омогућава да снимимо комплетан *HTTP* саобраћај. Биће нам доступни заглавља захтева, тела захтева, заглавља одговора и тела одговора.

2.2. Правила

Све у *ModSecurity*-ју се врти око две ствари: конфигурације и правила. Конфигурација говори *ModSecurity*-ју како да обради податке које види, а правила одлучују шта ће се радити са обрађеним подацима. Правила се пишу у формату [12]:

SecRule VARIABLES OPERATOR ACTION

Три дела имају следећа значења:

1. *VARIABLES* - део који говори *ModSecurity*-ју где да гледа. *ARGS* променљива значи да се гледају сви параметри захтева
2. *OPERATOR* део говори *ModSecurity*-ју како да гледа. Нпр. регуларни израз који упоређујемо са *ARGS*
3. *ACTIONS* део говори *ModSecurity*-ју шта да ради уколико дође до поклапања *ARGS*-а и *OPERATOR*-а

ModSecurity је такође потребно конфигурисати спрам потребе апликације. На слици 2 су приказане конфигурационе директиве *ModSecurity*-ја.

Directive	Description
SecDataDir	Sets the folder for persistent storage
SecRequestBodyAccess	Controls request body buffering
SecRequestBodyInMemoryLimit	Sets the size of the per-request memory buffer
SecRequestBodyLimit	Sets the maximum request body size ModSecurity will accept
SecRequestBodyLimitAction	Controls what happens once the request body limit is reached
SecRequestBodyNoFilesLimit	Sets the maximum request body size, excluding uploaded files
SecResponseBodyAccess	Controls response body buffering
SecResponseBodyLimit	Specifies the response body buffering limit
SecResponseBodyLimitAction	Controls what happens once the response body limit is reached
SecResponseBodyMimeType	Specifies a list of response body MIME types to inspect
SecResponseBodyMimeTypesClear	Clears the list of response body MIME types
SecRuleEngine	Controls the operation of the rule engine
SecTmpDir	Sets the folder for temporary files
SecUploadDir	Sets the folder in which intercepted files will be stored
SecUploadFileLimit	Set the maximum number of file uploads processed in a multipart POST
SecUploadKeepFiles	Controls whether the uploaded files will be kept after the transaction is processed

Слика 2. *Конфигурационе директиве ModSecurity*-ја

ModSecurity има главни прекидач - директиву *SecRuleEngine*. Ако ова директива има вредност *On*, тада се правила обрађују, док уколико је *Off* онда не. Такође може имати вредност *DetectionOnly*, у овом режиму правила се обрађују, али се не врше модификације у саобраћају.

3. ВЕРИФИКАЦИЈА MODSECURITY WAF-A

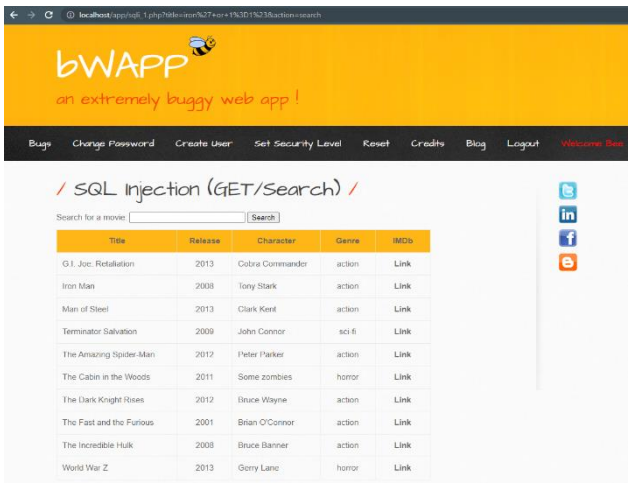
У овом поглављу биће приказано како функционише *ModSecurity* на примеру *bWAPP* (скраћено од *buggy web application*) апликације.

Правила се налазе унутар *.conf* фајлова. Ових фајлова иницијално има око 30. Правила која у префиксу имају реч *REQUEST* користе се за проверу захтева, док правила која проверавају одговор у префиксу имају реч *RESPONSE*.

3.1. Демонстрација правила

Демонстрација правила биће приказана на примеру покушаја *SQL Injection* напада.

Уколико искључимо *ModSecurity* и унесемо у *Search for a movie* поље *SQL Injection* команду - *iron' or 1=1#* у случају тестне апликације која је подложна овим рањивостима добили бисмо резултат те команде (слика 3).



Слика 3. *Резултат извршења SQL Injection команде*

У наведеном примеру видимо да је команда довела до тога да се прикажу сви филмови који постоје у бази што је сигурносни пропуст. Уколико укључимо *ModSecurity* и поново покушамо да извршимо исту команду добићемо *403 Forbidden* грешку што је приказано на слици 4.

Слика 4. *Покушај извршавања SQL injection команде са укљученим ModSecurity-јем*

Правило које се активирало је из фајла *REQUEST-942-APPLICATION-ATTACK-SQLI.conf,* а до испуњења услова за ово правило је дошло јер се аргумент захтева *title* поклопио са оператором *@detectSQLi* који користи *Libinjection*[1] за детекцију *SQLi* напада.

3.2. Перформансе

Перформансе су важан показатељ ефикасности *WAF*-ова, које представљају способност *firewall*-а да рукује мрежним захтевима.

У овом делу биће приказани резултати тестирања перформанси *bWAPP* апликације. Перформансе су мерене употребом *Apache JMeter* алата.

За мерење перформанси узето је у обзир више сценарија као што су покретање апликације без *ModSecurity*-ја, покретања са *ModSecurity*-јем и свим правилима омогућеним, покретање само са правилима везаним за проверавање одговора на захтев, покретање само са правилима везаним за проверавање захтева и покретање са *ModSecurity*-јем, али без укључених правила. Такође размотрено је и како број захтева утиче на перформансе у опсегу од 0 до 1500 захтева. На графику 1 приказани су резултати тестирања.

[1] *Libinjection* је *open-source* C библиотека која детектује *injection* нападе употребом лексичке анализе [13]

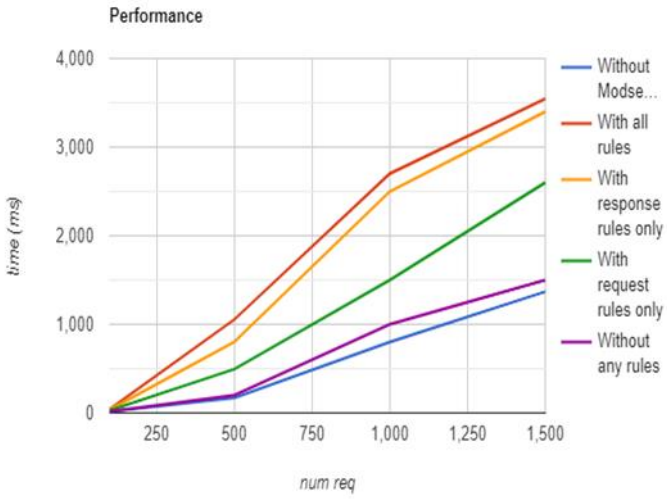


График 1 *Резултат тестирања перформанси апликације bWAPP*

Оно што се може закључити из графика јесте да правила која се користе за проверање захтева не утичу драстично на перформансе, док правила која се тичу проверавања одговора у циљу проналажења *data leakage*-а значајно утичу на перформансе. Из овога се може извући закључак да би требало прегледати правила која *ModSecurity* садржи и размотрити која правила су нам потребна за заштиту наше апликације и избацити она која нам нису потребна и на тај начин убрзати извршавање.

4. ЗАКЉУЧАК

У овом раду ближе је приказан *ModSecurity WAF*, који је формат правила на која се ослања као и од којих конфигурационих директива се састоји. У последњем поглављу је приказана и његова примена на примеру *bWAPP* апликације као и какав утицај има на перформансе.

Међутим, иако можемо добити пуно користи употребом *ModSecurity* -ја, и даље постоји неколико нерешених проблема. Један од њих је велики број *false positives* који се дешавају уколико се не поведе довољно рачуна око конфигурације. Неки даљи правци истраживања би били како смањити број *false positives*, а не нарушити безбедност апликације.

4. ЛИТЕРАТУРА

[1] Victor Clincy, Hossain Shahriar, „Web Application Firewall: Network Security Models and Configuration“, 42nd IEEE International Conference on Computer Software & Applications, 2018

[2] Abdul Razzaq, Ali Hur, Sidra Shahbaz, Muddassar Masood, H Farooq Ahmad, „Critical analysis on web application firewall solutions“, IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS), 2013.

[3] Web application firewall, https://www.cloudflare.com/learning/ddos/glossary/web-application-firewall-waf/

[4] Lakhno, V., A. Blozva, D. Kasatkin, V. Chubaievskyi, Y. Shestak, D. Tyshchenko, R. Brzhanov. "Experimental studies of the features of using waf to protect internal services in the zero trust structure." Journal of Theoretical and Applied Information Technology 100, no. 3 (2022).
[5] Khandelwal, Shashank, Parthiv Shah, Mr Kaushal Bhavsar, and Dr Savita Gandhi. "Frontline techniques to prevent web application vulnerability." Int. J. Advanced Research in Comput. Sci. Electron. Eng 2, no. 2 (2013): 208.
[6] Mac, Hieu, Dung Truong, Lam Nguyen, Hoa Nguyen, Hai Anh Tran, Duc Tran. "Detecting attacks on web applications using autoencoder." In Proceedings of the ninth international symposium on information and communication technology, pp. 416-421., 2018
[7]ModSecurity, https://en.wikipedia.org/wiki/ModSecurity
[8] Modsecurity-apache, SpiderLabs, https://github.com/SpiderLabs/ModSecurity-apache
[9] Orlando, Kyle Richard. "Automating Virtual Patching via Application Security Testing Tools." Master's thesis, NTNU, 2021
[10] ModSecurity-apache, https://tahir.pro/ModSecurity-apache/
[11] Jeichande, Dauto Ussene. "Redundant firewalls for web applications." PhD diss., 2016
[12] Ahmad, Ali, Zahid Anwar, Ali Hur, Hafiz Farooq Ahmad. "Formal reasoning of web application Firewall rules through ontological modeling." In 2012 15th International Multitopic Conference (INMIC), pp. 230-237. IEEE, 2012.
[13] Ashlam, Ahmed Abadulla, Atta Badii, and Frederic Stahl. "A Novel Approach Exploiting Machine Learning to Detect SQLi Attacks." In 2022 5th International Conference on Advanced Systems and Emergent Technologies (IC_ASET), pp. 513-517. IEEE, 2022

Кратка биографија:

Леона Недељковић рођена је 29.08.1997. године у Новом Саду. Основне академске студије завршила је на факултету техничких наука 2020. године. Мастер рад из области Електротехнике и рачунарства - Софтверско инжењерство и информационе технологије одбранила је 2022. године.