



# An Introduction to ModSecurity and the OWASP Core Rule Set

*(OWASP Hamburg)*

**Christian Folini / @ChrFolini**

Intro to ModSecurity and CRS - OWASP Hamburg 2021-04-14





# Safety Belts

Baseline / 1<sup>st</sup> Line of Defense

# Boring Bio

- **Christian Folini /  @ChrFolini**
- **Security Engineer at netnea in Switzerland**
- **Author, teacher and speaker**
- **OWASP CRS project Co-Lead**

## MODSECURITY HANDBOOK

The Complete Guide to the Popular  
Open Source Web Application Firewall



Christian Folini  
Ivan Ristić



# Plan for Today

- **What is a WAF?**
- **What is ModSecurity?**
- **What is Core Rule Set?**
- **Demo**
- **Key concepts**
- **Rules**
- **False Positives**



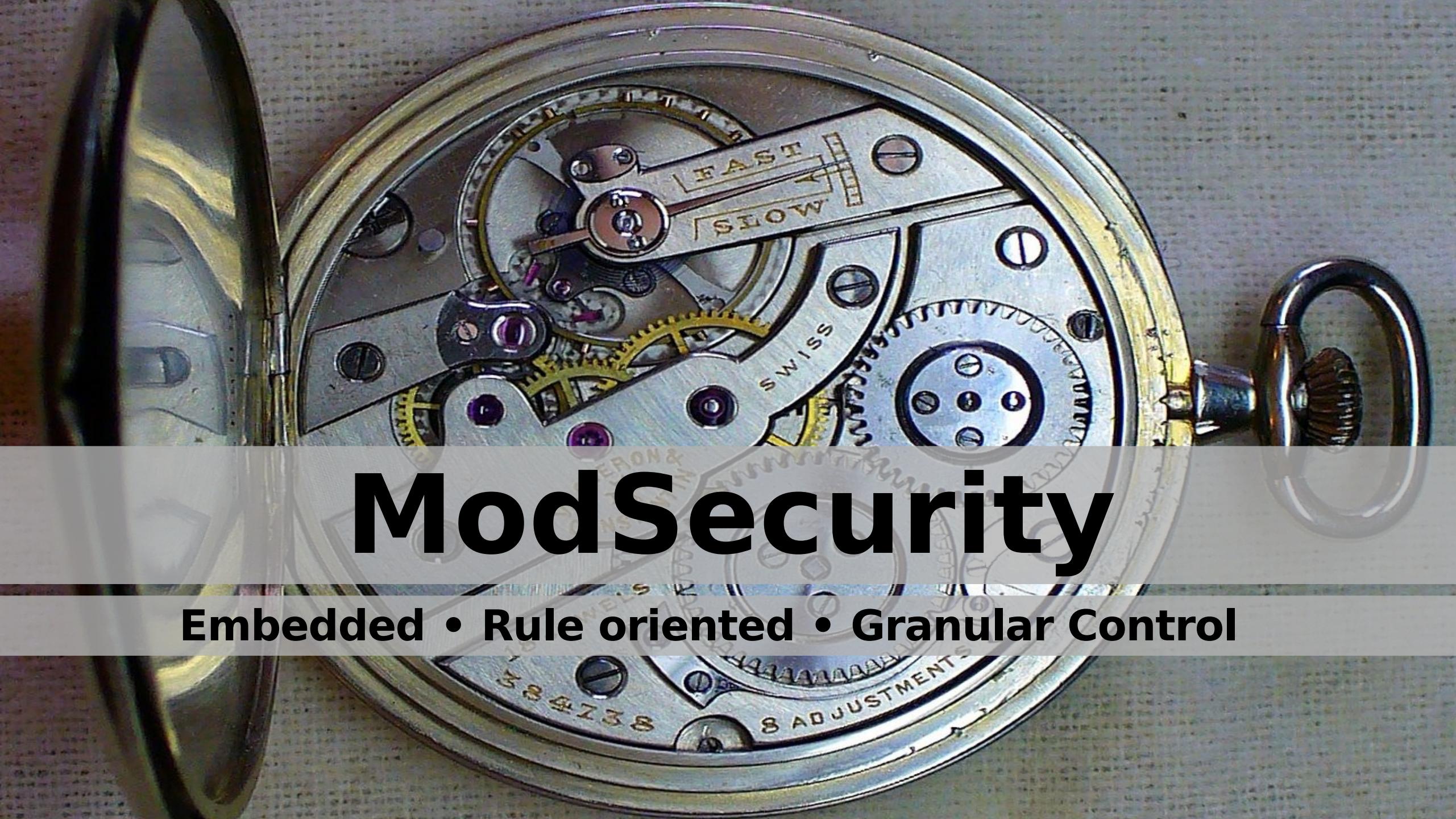
**CRS**  
THE 1<sup>ST</sup> LINE OF DEFENSE





# Web Application Firewalls

Complex • Overwhelming • Rarely Functional



# ModSecurity

**Embedded • Rule oriented • Granular Control**



**CRS**

OWASP  
ModSecurity  
Core Rule Set

THE 1<sup>ST</sup> LINE OF DEFENSE

INCLUDES  
DR FOLINI'S  
PARANOIA MODE



BASED UPON A TRUE STORY!

# CRS3

OWASP ModSecurity Core Rule Set v3.0

DIRECTED BY

CHAIM SANDERS

STARRING

WALTER HOP AS REGEX WIZARD, CHAIM SANDERS

ORIGINAL IDEA BY OFER SHEZAF AND RYAN BARNETT ALSO STARRING CHRISTIAN FOLINI, FRANZiska BÜHLER, @EMPHAZER, RYAN BARNETT, FELIPE ZIMMERLE,  
MANUEL LEOS, VLADIMIR IVANOV, CHRISTIAN PERON, @YGREK, @TOBY78, @JAMUSE, MATT KOCH, RICHIM HOFFMANN, MAZIN AHMED, NOËL ZINDEL



Smart security on demand  
COMING SOON TO A SERVER NEAR YOU!



OWASP  
Open Web Application  
Security Project

# Demo Time (Installation)

**Clone the repository (or download latest release):**

```
$> git clone https://github.com/coreruleset/coreruleset
```

**Copy the example config:**

```
$> cp crs-setup.conf.example crs-setup.conf
```

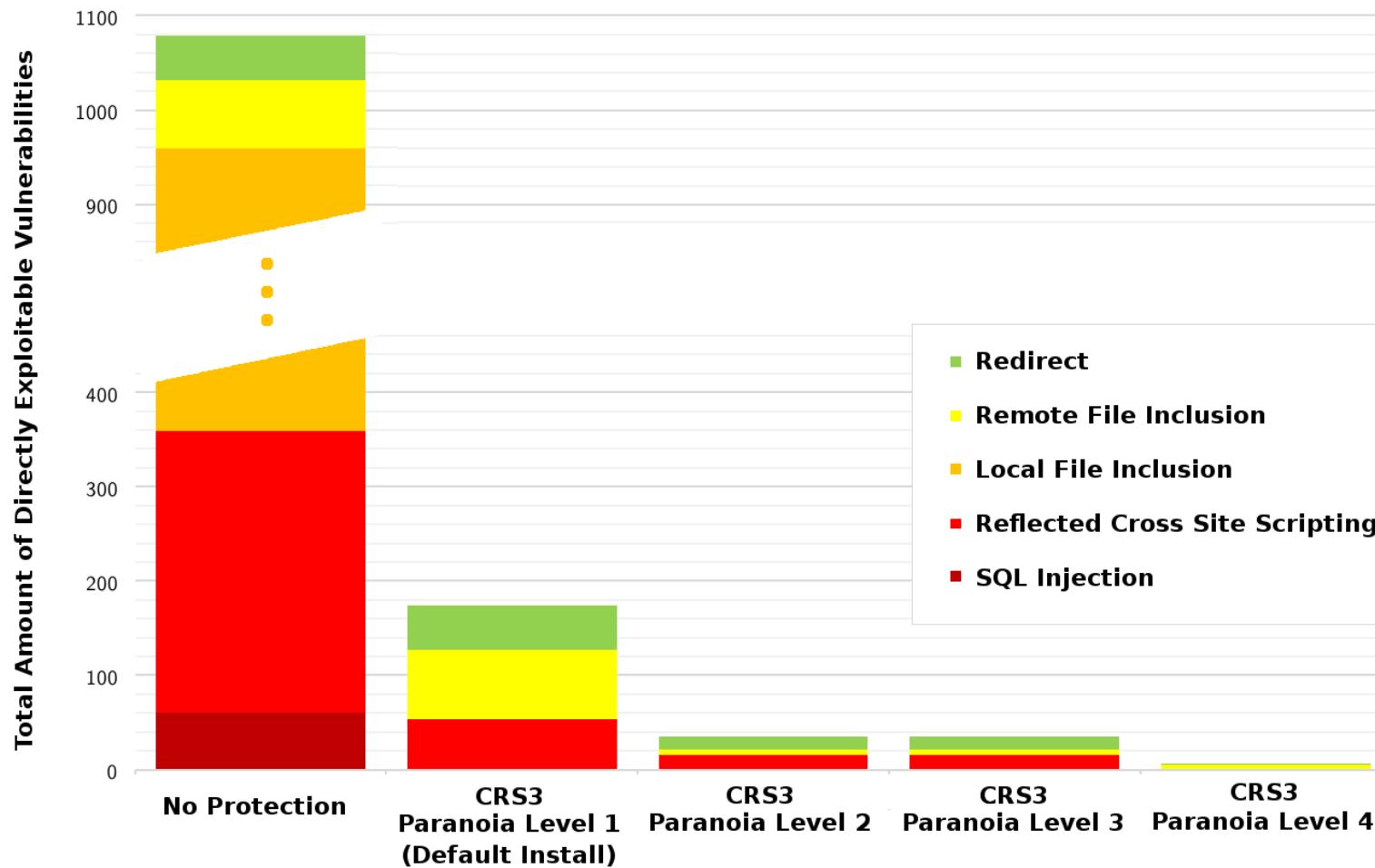
**Include in server config (depending on path):**

Include /path-to-owasp-crs/crs-setup.conf

Include /path-to-owasp-crs/rules/\*.conf



# Burp vs. OWASP ModSecurity Core Rule Set 3.0



## CRS3 Default Install

Redir.: 0%  
RFI: 0%  
**LFI: -100%**  
**XSS: -82%**  
**SQLi: -100%**

Research based on  
4.5M Burp requests.



# Paranoia Levels

## Paranoia Level 1: Minimal number of false positives

*Baseline protection*

## Paranoia Level 2: More rules, some false positives

*Real data in the service*

## Paranoia Level 3: Specialized rules, more FPs

*Online banking level security*

## Paranoia Level 4: Crazy rules, many FPs

*Nuclear power plant level security*



# Important Groups of Rules

## Request Rules

REQUEST-910-IP-REPUTATION.conf  
REQUEST-911-METHOD-ENFORCEMENT.conf  
REQUEST-912-DOS-PROTECTION.conf  
REQUEST-913-SCANNER-DETECTION.conf  
REQUEST-920-PROTOCOL-ENFORCEMENT.conf  
REQUEST-921-PROTOCOL-ATTACK.conf

REQUEST-930-APPLICATION-ATTACK-LFI.conf  
REQUEST-931-APPLICATION-ATTACK-RFI.conf  
REQUEST-932-APPLICATION-ATTACK-RCE.conf  
REQUEST-933-APPLICATION-ATTACK-PHP.conf  
REQUEST-941-APPLICATION-ATTACK-XSS.conf  
REQUEST-942-APPLICATION-ATTACK-SQLI.conf  
REQUEST-943-APPLICATION-ATTACK-SESS-FIX.conf  
REQUEST-944-APPLICATION-ATTACK-JAVA.conf

REQUEST-949-BLOCKING-EVALUATION.conf



**CRS**

THE 1<sup>ST</sup> LINE OF DEFENSE



# Important Groups of Rules

## Response Rules

**RESPONSE-950-DATA-LEAKAGES.conf**

**RESPONSE-951-DATA-LEAKAGES-SQL.conf**

**RESPONSE-952-DATA-LEAKAGES-JAVA.conf**

**RESPONSE-953-DATA-LEAKAGES-PHP.conf**

**RESPONSE-954-DATA-LEAKAGES-IIS.conf**

**RESPONSE-959-BLOCKING-EVALUATION.conf**



**CRS**

THE 1<sup>ST</sup> LINE OF DEFENSE



# Paranoia Level

## Example: Protocol Enforcement Rules

**Paranoia Level 1:** **31 Rules**

**Paranoia Level 2:** **7 Rules**

**Paranoia Level 3:** **1 Rules**

**Paranoia Level 4:** **4 Rules**



**CRS**  
THE 1<sup>ST</sup> LINE OF DEFENSE



# Stricter Siblings

## Example: Byte Range Enforcement

**Paranoia Level 1:**

**Rule 920270: Full ASCII range without null character**

**Paranoia Level 2:**

**Rule 920271: Full visible ASCII range, tab, newline**

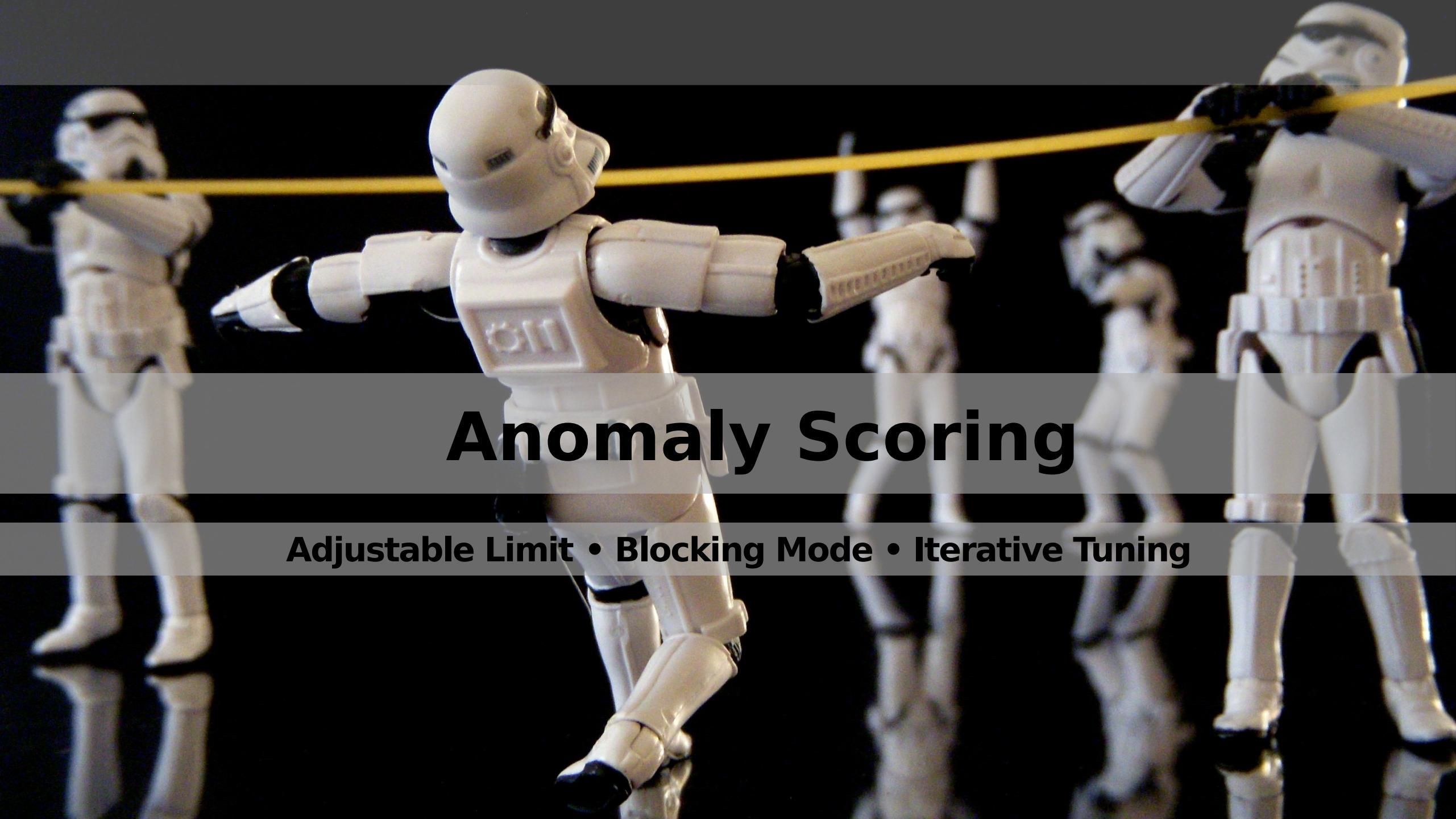
**Paranoia Level 3:**

**Rule 920272: Visible lower ASCII range without %**

**Paranoia Level 4:**

**Rule 920273: A-Z a-z 0-9 = - \_ . , : &**





# Anomaly Scoring

Adjustable Limit • Blocking Mode • Iterative Tuning

# **Sampling Mode**

**Easing into CRS adoption / limit the impact**

- **Define a sampling rate of n**
- **Only n% of the requests are being funneled into CRS3**
- **100% - n% of requests bypass CRS3**
- **Monitor performance and fix problems**
- **Slowly raise n in an iterative way until it reaches 100%**



What would you like to learn about today? Search our documentation...**Getting started**

- ▶ Basics
- ▶ Domains & Origins
- ▶ Performance

**Configuration**

- ▶ Basics
- ▶ Conditions
- ▶ Dictionaries
- ▶ Domains & Origins
- ▶ Request settings
- ▶ Cache settings
- ▶ Headers
- ▶ Responses
- ▶ Performance
- ▶ Purging
- ▶ Custom VCL
- ▶ Image optimization
- ▶ Video

**Security**

- ▶ Access Control Lists
- ▶ Monitoring and testing
- ▶ Securing communications
- ▶ Security measures
- ▶ TLS
- ▶ Web Application Firewall
- [About the Fastly WAF dashboard](#)
- [Creating a custom WAF error page](#)

[Home](#) > [Guides](#) > [Security](#)

# Fastly WAF rule set updates and maintenance

Last updated June 26, 2019

Fastly provides rule set updates to the [Fastly WAF](#) in a prompt manner to help protect customers against attacks.

For OWASP and Trustwave rules changes we use the following process:

1. We regularly review the rule changes as they happen in both the OWASP Core Rule Set and the Trustwave Rule Set.
2. We translate the rules into [Varnish Configuration Language \(VCL\)](#) to run inside our cache nodes.
3. We test the rules in our platform to ensure they perform adequately. We try to maximize performance and rule efficacy while reducing false positives.
4. We correct bugs, if any are found.
5. We propagate the rule set changes to our platform worldwide.
6. Finally, we will provide customers with a notification and [instructions on how to make rule updates](#).

**IMPORTANT:** This information is part of a limited availability release. For more information, see our [product and feature lifecycle](#) descriptions.



AWS Services Resource Groups

AWS WAF > Web ACLs > Create web ACL

Step 1  
Describe web ACL and associate AWS resources

Step 2  
Add rules and rule groups: Add managed rule groups

Step 3  
Set rule priority

Step 4  
Configure metrics

Step 5  
Review and create web ACL

## Add managed rule groups Info

Managed rule groups are created and maintained for you by AWS and AWS Marketplace sellers.

**AWS managed rule groups**

Name	Capacity	Action
<b>Admin protection</b> Contains rules that allow you to block external access to exposed admin pages. This may be useful if you are running third-party software or would like to reduce the risk of a malicious actor gaining administrative access to your application.	100	<input type="radio"/> Add to web ACL <input type="radio"/> Set rules action to count
<b>Amazon IP reputation list</b> This group contains rules that are based on Amazon threat intelligence. This is useful if you would like to block sources associated with bots or other threats.	25	<input type="radio"/> Add to web ACL <input type="radio"/> Set rules action to count
<b>Core rule set</b> Contains rules that are generally applicable to web applications. This provides protection against exploitation of a wide range of vulnerabilities, including those described in OWASP publications and common Common Vulnerabilities and Exposures (CVE).	700	<input type="radio"/> Add to web ACL <input type="radio"/> Set rules action to count
<b>Known bad inputs</b> Contains rules that allow you to block request patterns that are known to be invalid and are associated with exploitation or discovery of vulnerabilities. This can help reduce the risk of a malicious actor discovering a vulnerable application.	200	<input type="radio"/> Add to web ACL <input type="radio"/> Set rules action to count
<b>Linux operating system</b> Contains rules that block request patterns associated with exploitation of vulnerabilities specific to Linux, including LFI attacks. This can help prevent attacks that expose file contents or execute code for which the attacker should not have had access.	200	<input type="radio"/> Add to web ACL <input type="radio"/> Set rules action to count
<b>PHP application</b> Contains rules that block request patterns associated with exploiting vulnerabilities specific to the use of PHP, including injection of unsafe PHP functions. This can help prevent exploits that allow an attacker to remotely execute code or commands.	100	<input type="radio"/> Add to web ACL <input type="radio"/> Set rules action to count

Close



# CRS

THE 1<sup>ST</sup> LINE OF DEFENSE



 Microsoft Azure

Contact Sales: 1-800-867-1389  Search  Portal

Overview Solutions Products  Documentation Pricing Training Marketplace  Partners  More  Free account >

Azure / Web Application Firewall

Learn Azure at your own pace  

 Filter by title

Azure Web Application Firewall Documentation

✓ Overview

- What is Azure Web Application Firewall?

✓ Application Gateway

- Web Application Firewall on Application Gateway**
- What's new
- Front Door

Tutorials

Samples

Concepts

How-to guides

Troubleshoot

Reference

Resources

# Azure Web Application Firewall on Azure Application Gateway

11/14/2019 • 8 minutes to read • 

## In this article

[Benefits](#)

[Features](#)

[WAF Policy](#)

[Application Gateway WAF SKU pricing](#)

[Next steps](#)

Azure Web Application Firewall (WAF) on Azure Application Gateway provides centralized protection of your web applications from common exploits and vulnerabilities. Web applications are increasingly targeted by malicious attacks that exploit commonly known vulnerabilities. SQL injection and cross-site scripting are among the most common attacks.

WAF on Application Gateway is based on [Core Rule Set \(CRS\)](#) 3.1, 3.0, or 2.2.9 from the Open Web Application Security Project (OWASP). The WAF automatically updates to include protection against new vulnerabilities, with no additional configuration needed.



**CRS**

THE 1<sup>ST</sup> LINE OF DEFENSE



@ChrFolini Intro to ModSecurity and CRS - OWASP Hamburg 2021-04-14

Protection Rules

Rules Recommendations Rule Settings

Rules ⓘ

Actions ▾

<input type="checkbox"/>	Rule ID	Protection Rule	Action
<input type="checkbox"/>	941340	Cross-Site Scripting (XSS) Attempt: XSS Filters from Internet Explorer Cross-Site Scripting (XSS) Attempt: XSS Filters from IE	Block
<input type="checkbox"/>	941330	Cross-Site Scripting (XSS) Attempt: XSS Filters from Internet Explorer Cross-Site Scripting (XSS) Attempt: XSS Filters from IE	Block
<input type="checkbox"/>	941320	Cross-Site Scripting (XSS) Attempt: HTML Tag Handler Cross-Site Scripting (XSS) Attempt: HTML Tag Handler	Block
<input type="checkbox"/>	941150	Cross-Site Scripting (XSS) Attempt: XSS Filters - Category 5, HTML attributes - src, style and href	Block
<input type="checkbox"/>	941101	Cross-Site Scripting (XSS) Attempt: SS Attack Detected via libinjection Cross-Site Scripting (XSS) Attempt: SS Attack Detected via libinjection	Block
<input type="checkbox"/>	941350	Cross-Site Scripting (XSS) Attempt: UTF-7 encoding XSS filter evasion for IE Cross-Site Scripting (XSS) Attempt: UTF-7 encoding XSS filter evasion for IE	Block
Cross-Site Scripting (XSS) Attempt: US-ASCII encoding bypass listed on XSS filter evasion			

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.



# CRS

THE 1<sup>ST</sup> LINE OF DEFENSE

## Tightly Integrated into the Oracle Cloud Infrastructure Console

@ChrFolini Intro to ModSecurity and CRS - OWASP Hamburg 2021-04-14



# Cloudflare Support

 Search our knowledge base...

Popular topics: I'm under DDoS, Using Page Rules, Restoring IPs on server logs

Cloudflare Support > Firewall > Managed Rules - Web Application Firewall (WAF)

> Managing the OWASP rule set in the WAF

## Managing the OWASP rule set in the WAF

Cloudflare Support (May 14, 2019 11:46)

Follow

With Cloudflare Web Application Firewall (WAF), you can control the level of sensitivity to apply and the action to take when a threat is detected, as determined by the OWASP rule set.



Cloudflare **Web Application Firewall** (WAF) is available to customers in the Pro plan and above. To learn more about our plans, visit [Cloudflare Pricing](#).

### Understand OWASP rule set sensitivity and action

When responding to a potential web application threat, Cloudflare triggers actions based on a threat score that is assigned to each incoming request. When a request triggers an OWASP rule, that rule increases the request's overall threat score. Some rules increase the score more than others.

Cloudflare provides three sensitivity settings for the OWASP rule set: high, medium, and low. The table

### ⚠ Minor Service Outage

[Detailed system status](#) >

#### Related articles

[Cloudflare Logs \(formerly ELS\)](#)

[Hardening WordPress Security with the Cloudflare Firewall](#)

[Configuring Rate Limiting in the Cloudflare Dashboard](#)

[Cloudflare Rate Limiting](#)

[Understanding and Configuring DNSSEC in Cloudflare DNS](#)

#### Connect with the Cloudflare Community

[Get answers](#)



# CRS

THE 1<sup>ST</sup> LINE OF DEFENSE



@ChrFolini Intro to ModSecurity and CRS - OWASP Hamburg 2021-04-14

January 29, 2019

## Running a multi-tenant WAF at the edge

By Reed Morrison, Software Developer

Web Application Firewalls (WAFs) are a critical layer in modern web security, providing a website's first line of defense against vulnerabilities. WAFs can be used to defend against and notify on attempted exploits, allowing for mitigations faster than organizations can patch vulnerable software. For a global CDN, this functionality must be implemented in a way that is sensitive to performance, providing response times on the order of milliseconds. When we first introduced a WAF engine to the Verizon Digital Media Services stack three years ago, we selected the [ModSecurity Rules Engine](#), which we found to be first-rate for individual WAF use cases. Furthermore, ModSecurity's support of the [OWASP Core Rule Set \(CRS\)](#), powerful rule language, and API access to the HTTP traffic stream in real time offered significant flexibility.

### Enter waflz

However, as the number of customers using the WAF increased, we experienced performance and resource bottlenecks. ModSecurity's dense ruleset propagated across every customer instance drove memory and CPU utilization up across our network, increasing operational costs. Additionally, testing and deploying new rules was difficult because the rule language was often unwieldy and difficult to write and parse. These issues, along with development complexity with the existing ModSecurity library, led to the development of [waflz](#), an open source WAF engine, published under the [Apache 2.0 license](#).

For Verizon Digital Media Services, waflz is a significant improvement on ModSecurity because:

- It consumes less memory.
- Offers better performance.
- Is API-driven.

Waflz supports a subset of ModSecurity capabilities, the OWASP Core rulesets 2.x and 3.x, and several third-party rulesets.

**CRS****THE 1<sup>ST</sup> LINE OF DEFENSE**

# False Positives

**False Positives are expected from PL2**

- FPs are fought with rule exclusions
- Tutorials at <https://www.netnea.com>
- Get cheatsheet from Netnea
- Please report FPs at PL1 (github)



## RULE EXCLUSIONS

### ENTIRE RULES

#### STARTUP TIME

WHEN STARTING SERVER   WHEN RELOADING SERVER  
PLACE AFTER CRS INCLUDE

**SecRuleRemoveById**  
SecRuleRemoveByTag

SecRuleRemoveById 942100,...  
SecRuleRemoveByTag "attack-sql"

#### RUN TIME

WHEN EXAMINING A REQUEST   PLACE BEFORE CRS INCLUDE

**ctl:ruleRemoveById**  
ctl:ruleRemoveByTag

"...,ctl:ruleRemoveById:920300"  
"...,ctl:ruleRemoveByTag:attack-sql"

### PARAMETER IN RULES

#### STARTUP TIME

WHEN STARTING SERVER   WHEN RELOADING SERVER  
PLACE AFTER CRS INCLUDE

**SecRuleUpdateTargetById**  
SecRuleUpdateTargetByTag

SecRuleUpdateTargetById 942100 !ARGS:password  
SecRuleUpdateTargetByTag "attack-sql" !ARGS:password

#### RUN TIME

WHEN EXAMINING A REQUEST   PLACE BEFORE CRS INCLUDE

**ctl:ruleRemoveTargetById**  
ctl:ruleRemoveTargetByTag

"...,ctl:ruleRemoveTargetById:942100;ARGS:password"  
"...,ctl:ruleRemoveTargetByTag:attack-sql;ARGS:password"



# Apache / ModSecurity / CRS Tutorials

**<https://www.netnea.com/cms/apache-tutorials/>**

- Tutorial 1: Compiling Apache (Video Walk-Through)
- Tutorial 2: Configuring a Minimal Apache Web Server
- Tutorial 3: Configuring an Apache/PHP Application Server
- Tutorial 4: Enabling Encryption with SSL/TLS
- Tutorial 5: Extending and Analyzing the Access Log
- Tutorial 6: Embedding ModSecurity
- Tutorial 7: Including OWASP ModSecurity Core Rule Set
- Tutorial 8: Handling False Positives with the OWASP ModSecurity Core Rule Set
- Tutorial 9: Setting up a Reverse Proxy Server
- Tutorial 10: Efficiently Configuring and Debugging Apache and ModSecurity in the Shell
- Tutorial 11: Visualization of Apache / ModSecurity log information
- Tutorial 12: Capturing and Decrypting the Entire Traffic



# ModSecurity / CRS Courses

- Offered at <https://netnea.com>
- 1 seat to give away for free for next week, April 22 / 23

**US Time-Zone (15:00 - 23:00 CET)**



# Summary ModSecurity & CRS3

- **1<sup>st</sup> Line of Defense against web application attacks**
- **Generic set of blacklisting rules for WAFs**
- **Blocks 80% of web application attacks in the default installation (with a minimal number of FPs)**
- **Granular control over the behaviour down to the parameter level**

**More information at <https://coreruleset.org>**



# Questions and Answers, Contact

Contact:

[christian.folini@netnea.com](mailto:christian.folini@netnea.com)



@ChrFolini



**CRS**

THE 1<sup>ST</sup> LINE OF DEFENSE

