Security Summit

Cagliari, 18 settembre 2024

The rise and fall of ModSecurity and the Core Rule Set

(ovvero come evadere i WAF tramite degli attacchi adversarial)

Davide Ariu | CEO & Co-Founder

Pluribus One

whoami

PLURIBUS ONE CEO & Co-Founder

OWASP Italy Co-Chair

15 Years as academic researcher in Cybersecurity & AI
(Univ. of Cagliari, Georgia Tech)

www.apptake.eu **Project Coordinator**

Mainteiner of **UNBOXED APPSEC**
(http://davideariu.substack.com)

https://www.linkedin.com/in/davideariu/

Acknowledgments

http://apptake.eu

http://elsa-ai.eu

https://kinaitics.eu

KINAITICS has been funded by the European Union under Grant Agreement 101070176
ELSA has been funded by the European Union under Grant Agreement 101070617

APPTAKE has been funded under Grant Agreement No. 101128082 is supported by the European Cybersecurity Competence Centre

Acknowledgments

http://nerocybersecurity.eu

https://cybersuiteproject.eu

NERO has been funded under Grant Agreement No. 101127411 is supported by the European Cybersecurity Competence Centre

CYBERSUITE has been funded under Grant Agreement No. 101145861 is supported by the European Cybersecurity Competence Centre

Acknowledgments

“***ModSec-Learn: Boosting ModSecurity with Machine Learning***” - C. Scano, G. Floris , B. Montaruli, L. Demetrio, A. Valenza, L. Compagna, D. Ariu, L. Piras , D. Balzarotti, and B. Biggio - **DCAI - Salamanca 26th - 28th June, 2024**

“***Adversarial ModSecurity: Countering Adversarial SQL Injections with Robust Machine Learning***” – B. Montaruli, L. Demetrio, A. Valenza, L. Compagna, D. Ariu, L. Piras, D. Balzarotti, B. Biggio - **arXiv August 2023**

OWASP AppSec Lisbona 2024

Extended Version of this Talk

https://www.youtube.com/watch?v=LfQBIN6xYQY

Organization of this presentation

1. Introduction to WAFs and their detection mechanisms

2. Introduction to the OWASP CRS (key concepts)

3. **Original research results #1**: *boosting CRS performances with ML (**ModSec-Learn**)*

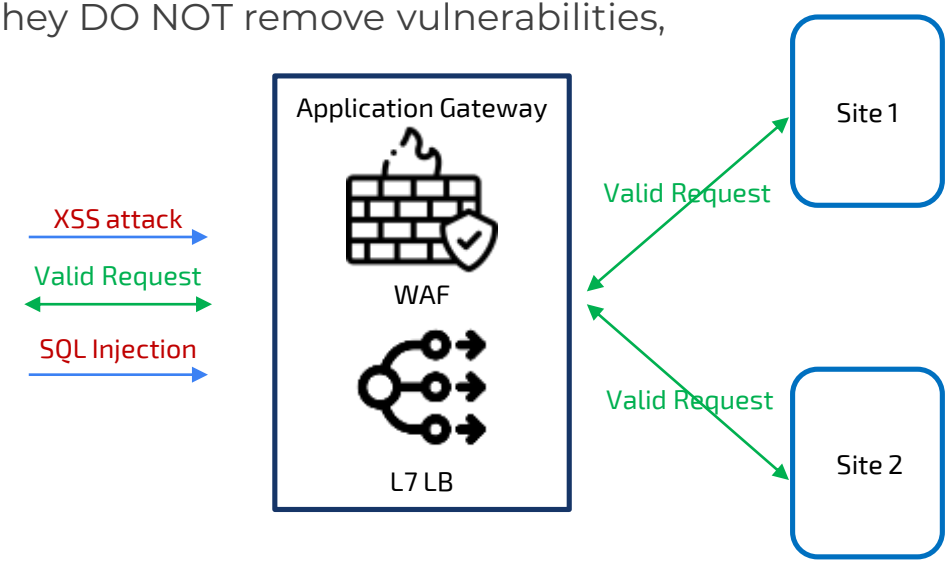
4. **Original research results #2**: making the CRS robust against adversarial attacks *(**AdvModSec**)*

Web Application Firewalls
Fundamentals

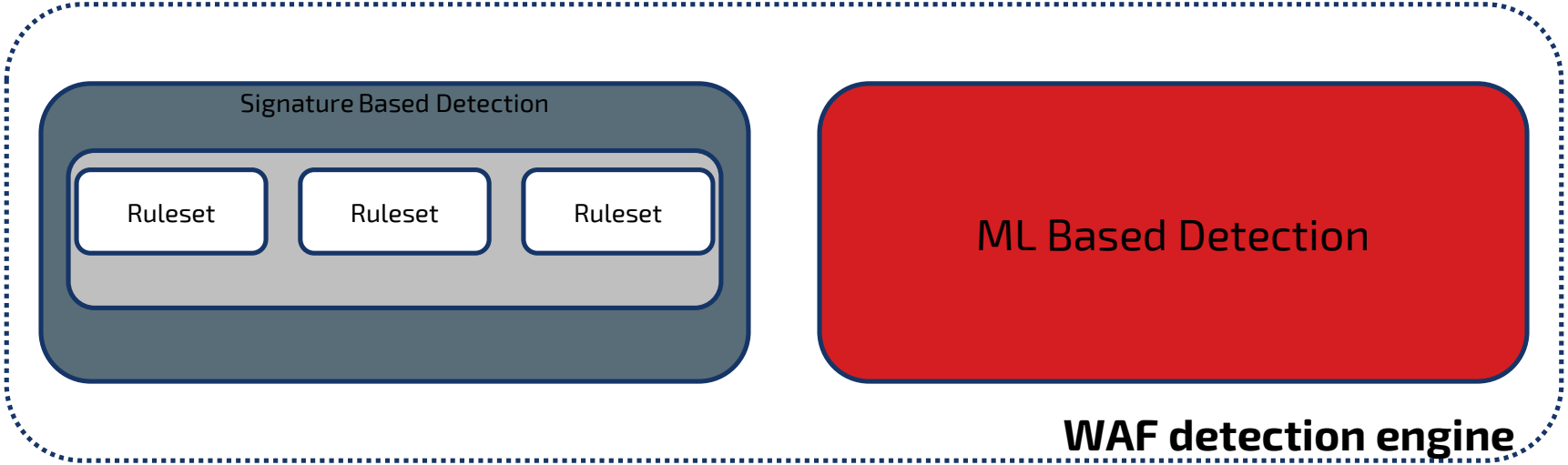
- Deployed “in front” of web applications to protect them from attacks

- WAFs are a quick and easy solution, but they DO NOT remove vulnerabilities, just hide them under the rug

- Very useful to “patch” applications, also block some “unexpected” attacks, but far from perfect



Components of a WAF Detection Engine



The *rulesets* are the elements actually responsible for the definition of the attacks

- It is basically a set of regEx applied at the HTTP protocol layer
- Can be applied both on the headers and the body of requests and responses

Key CRS Concepts

The pivotal role of the Core Rule Set

OWASP ModSecurity Core Rule Set
THE 1st LINE OF DEFENSE

WAF	Free plan	Free trial	Rules	ML services
Wallarm	✗	✓ (28 days)	Proprietary (OWASP Top 10 + API)	Wallarm AI Engine
CloudFlare	✓	✓ (30 days)	Proprietary (OWASP Top 10)	WAF-ML (only for SQL-i and XSS)
AWS	✗	✗ (PAYG)	AWS rules (CRS) or third-party (Fortinet, F5)	Amazon Lookout for Metrics (add-on service)
Azure	✗	✗ (200$ credit for 30 days)	OWASP CRS 3.2	Microsoft Sentinel (add-on service)
Google	✗	✗ (300$ credit)	OWASP CRS 3.3	Adaptive Protection (only DDoS)
Fortinet	✗	✓	Proprietary (OWASP Top 10 + API)	FortiWeb ML (Anomaly & bot detection)
F5	✗	✓ (30 days)	Proprietary (OWASP Top 10 + API)	NGINX App Protect DoS & Adaptive Violation Rating of WAF
Fastly	✗	✓	Proprietary (OWASP Top 10 + API)	Fastly SmartParse
Imperva	✗	✓ (30 days)	Proprietary (OWASP Top 10 + API)	Imperva Attack Analytics

The OWASP Core Rule Set

OWASP ModSecurity Core Rule Set
THE 1ST LINE OF DEFENSE

A *flagship* OWASP project

- The Core Rule Set (CRS) is a set of generic attack detection rules for use with ModSecurity, Coraza, or other compatible Web Application Firewalls.

Request Rules

```
REQUEST-905-COMMON-EXCEPTIONS.conf
REQUEST-911-METHOD-ENFORCEMENT.conf
REQUEST-913-SCANNER-DETECTION.conf
REQUEST-920-PROTOCOL-ENFORCEMENT.conf
REQUEST-921-PROTOCOL-ATTACK.conf
REQUEST-922-MULTIPART-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-934-APPLICATION-ATTACK-GENERIC.conf
REQUEST-941-APPLICATION-ATTACK-XSS.conf
REQUEST-942-APPLICATION-ATTACK-SQLI.conf
REQUEST-943-APPLICATION-ATTACK-SESSION-FIXATION.conf
REQUEST-944-APPLICATION-ATTACK-JAVA.conf
```

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-955-WEB-SHELLS.conf
RESPONSE-959-BLOCKING-EVALUATION.conf
```

Source: https://github.com/coreruleset/coreruleset/tree/main/rules

The OWASP Core Rule Set

Key concepts from the Core Rule Set will be recalled in the following slides

- Rules structures –> **Severity** (associated with every single rule)
- **Anomaly Scoring** (assigned to the requests/responses)
- Paranoia Level (used to select the set of rules)

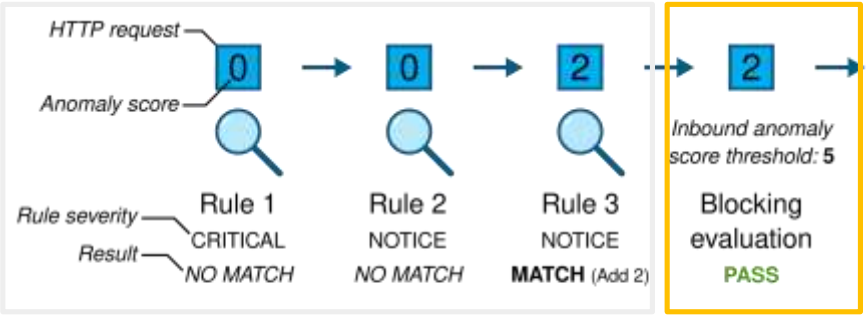
OWASP CRS – Rules structure

```
SecRule REQUEST_HEADERS:Content-Length "!@rx ^\d+$" \
    "id:920160,\
    phase:1,\
    block,\
    t:none,\
    msg:'Content-Length HTTP header is not numeric',\
    logdata:'%{MATCHED_VAR}',\
    tag:'application-multi',\
    tag:'language-multi',\
    tag:'platform-multi',\
    tag:'attack-protocol',\
    tag:'paranoia-level/1',\
    tag:'OWASP_CRS',\
    tag:'capec/1000/210/272',\
    ver:'OWASP_CRS/3.4.0-dev',\
    severity:'CRITICAL',\
    setvar:'tx.anomaly_score_pl1=+%{tx.critical_anomaly_score}'"
```

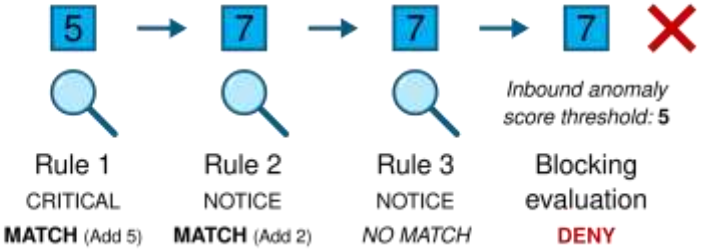
Severity Level	Default Anomaly Score
CRITICAL	5
ERROR	4
WARNING	3
NOTICE	2

OWASP CRS – Anomaly scoring

Collaborative detection



Delayed blocking



Execute all *request* rules

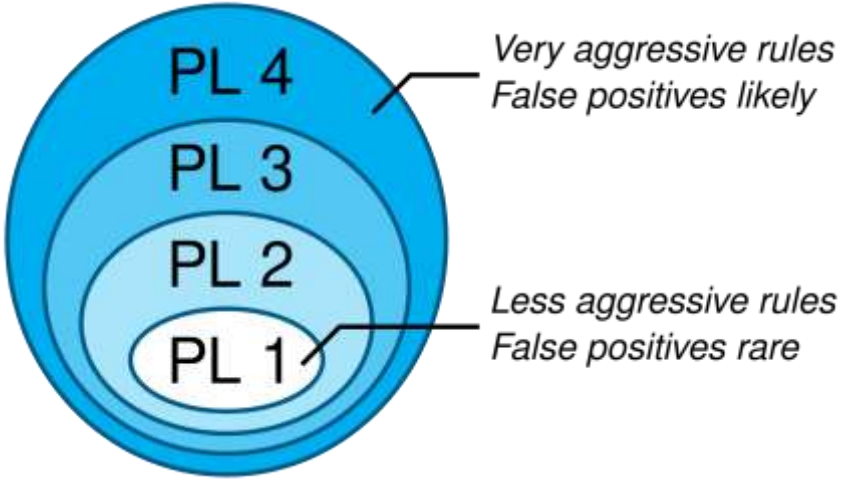
Make a blocking decision using the *inbound* anomaly score threshold

Execute all *response* rules

Make a blocking decision using the *outbound* anomaly score threshold

Source: https://coreruleset.org/docs/concepts/anomaly_scoring/

OWASP CRS – Paranoia Level

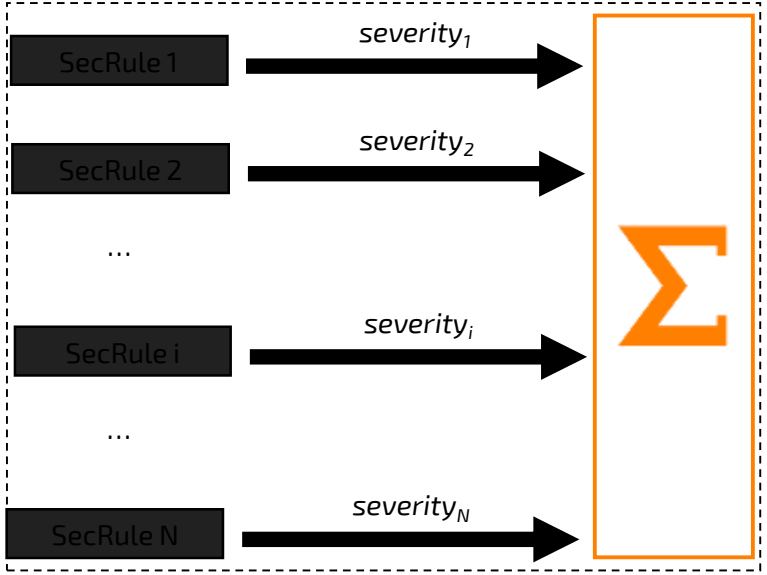


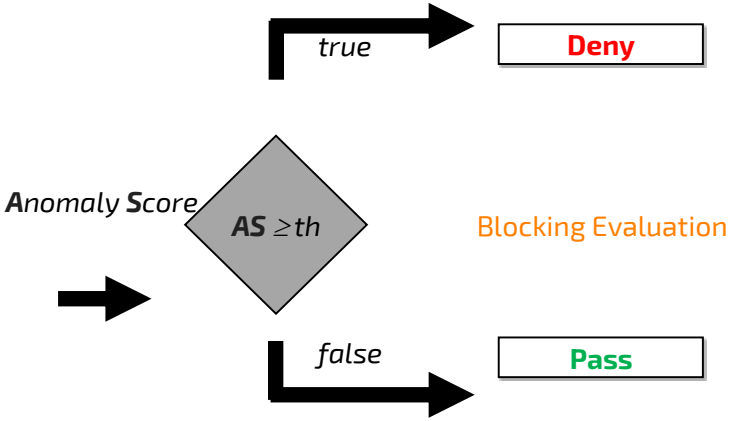
Paranoia	Description
1	Baseline security with a minimal need to tune away false positives.
2	Rules that are adequate when real user data is involved.
3	Online banking level security with lots of false positives.
4	Very strict rules that generate many false positives.

Source: https://coreruleset.org/docs/concepts/paranoia_levels/

A practical example of how the CRS works

```
SELECT * FROM
items WHERE owner
= 'wiley' AND
itemname = 'name'
OR 'a'='a';
```





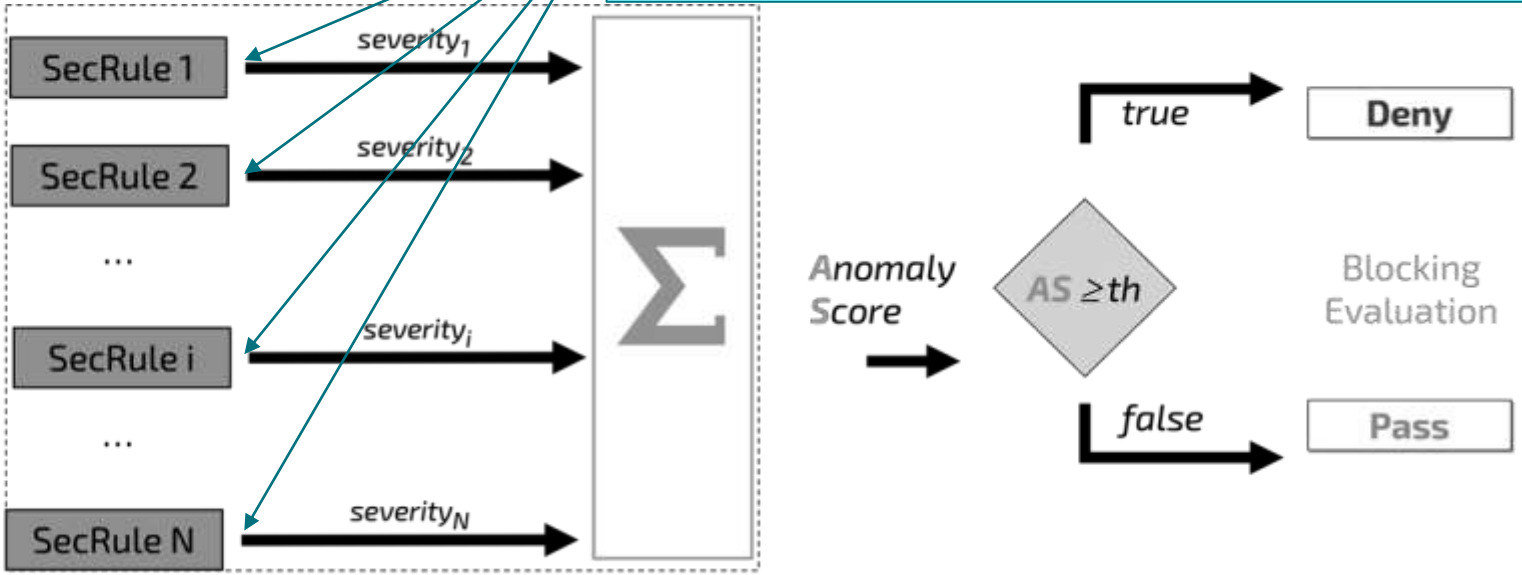
Issues with CRS performance tuning

#1 – Sets of rules used

The set of rules to use is selected a priori based on the paranoia level (+ exclusions)

- Whether a rule is relevant or not is not based on the traffic
- Redundancy among the rules is also possible

```
SELECT * FROM
items WHERE owner
= 'wiley' AND
itemname = 'name'
OR 'a'='a';
```



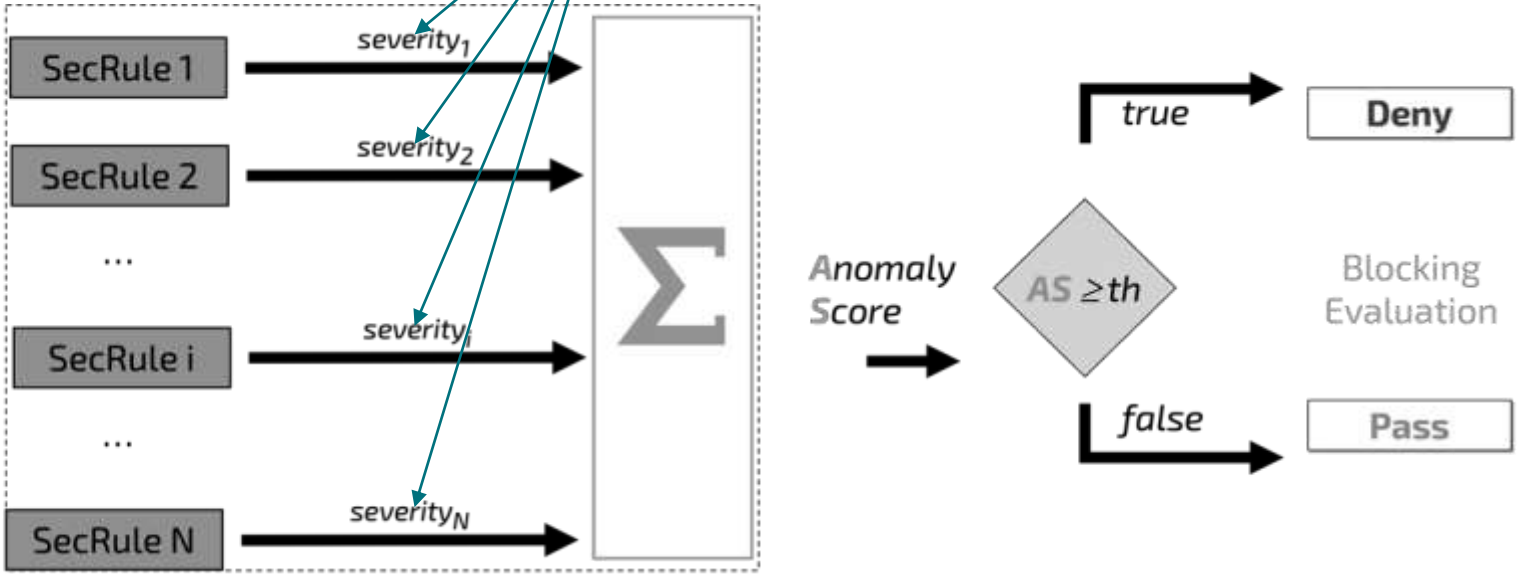
Issues with CRS performance tuning

#2 – Severity levels

The severity of each rule is purely heuristic

- Not direct correlation with the traffic incoming to the applications

```
SELECT * FROM
items WHERE owner
= 'wiley' AND
itemname = 'name'
OR 'a'='a';
```



Original Research Results #1

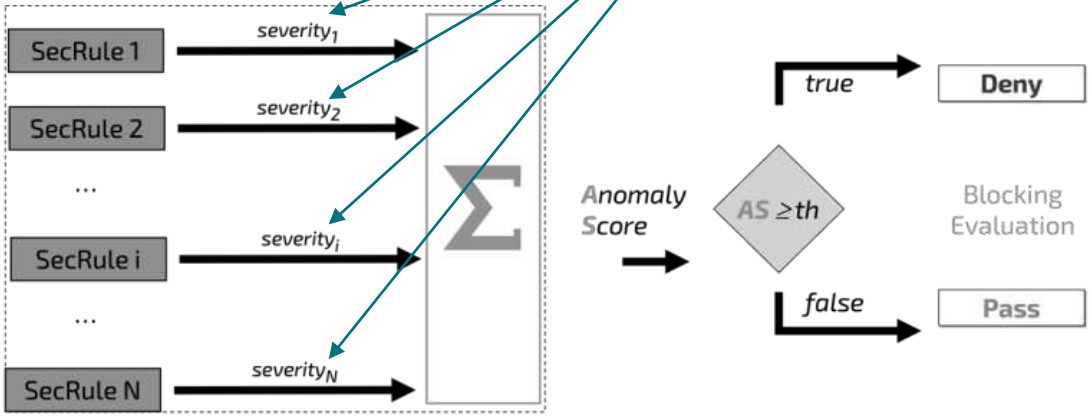
Boosting CRS performances with ML (MLModSec)

Bring ML into the CRS decision making process

Step#1 - Severity estimates

Based on the traffic (legitimate, malicious), it is possible to find an estimate for the rule *severity* values which weights more the rule that contribute significantly to the detection without generating false positives

```
SELECT * FROM
items WHERE owner
= 'wiley' AND
itemname = 'name'
OR 'a'='a';
```

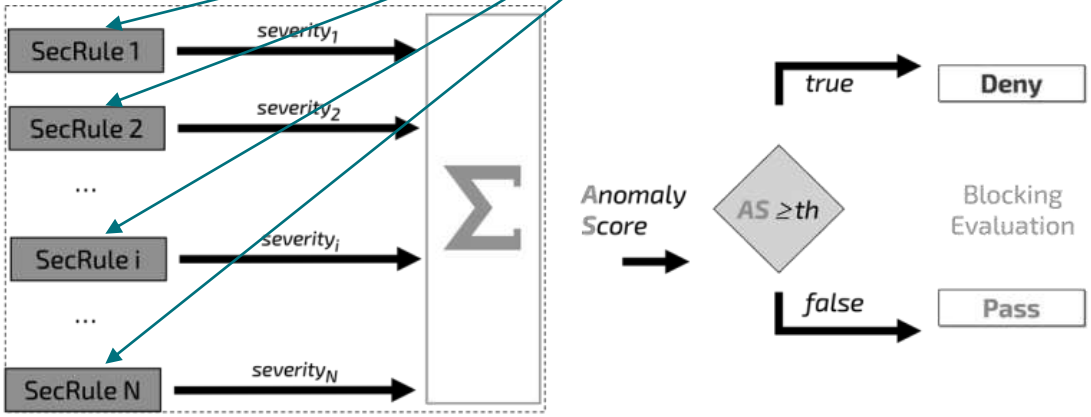


Bring ML into the CRS decision making process

Step#2 - Rules Selection

Based on the same process we are «selecting» the rules that are effectively contributing to the detection. Redundant or not necessary rules are pruned setting the corresponding severity to 0.

```
SELECT * FROM
items WHERE owner
= 'wiley' AND
itemname = 'name'
OR 'a'='a';
```

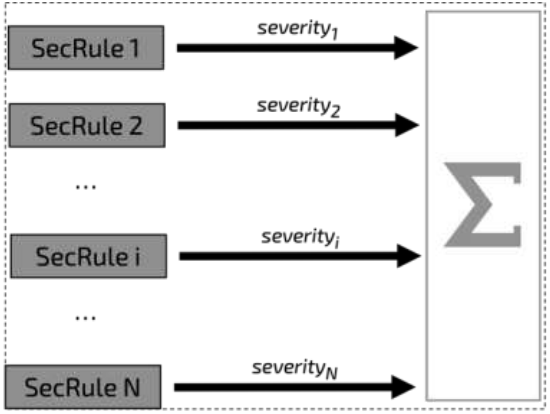


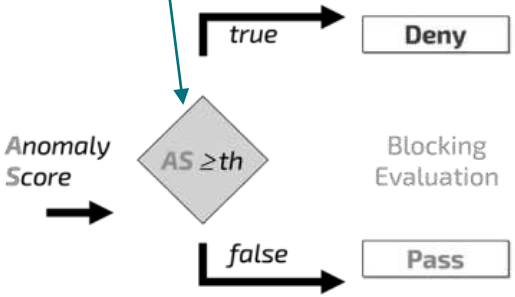
Bring ML into the CRS decision making process

Step#3 - Threshold estimate

The threshold is set based on a target value for detection and the false positive rate

```
SELECT * FROM
items WHERE owner
= 'wiley' AND
itemname = 'name'
OR 'a'='a';
```

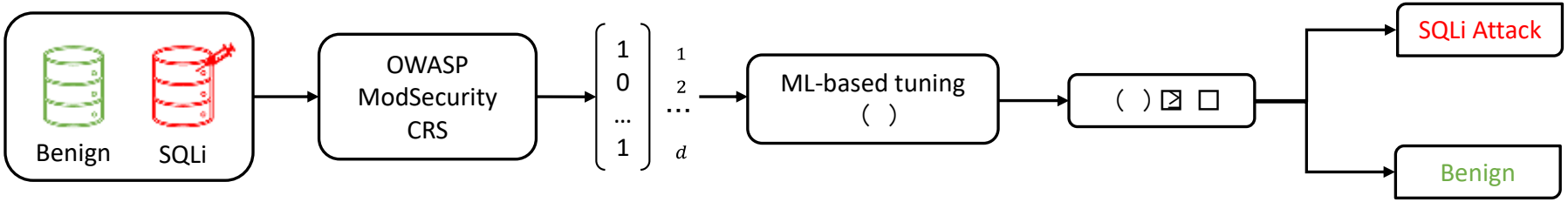




ModSec-Learn

Boosting ModSecurity with Machine Learning

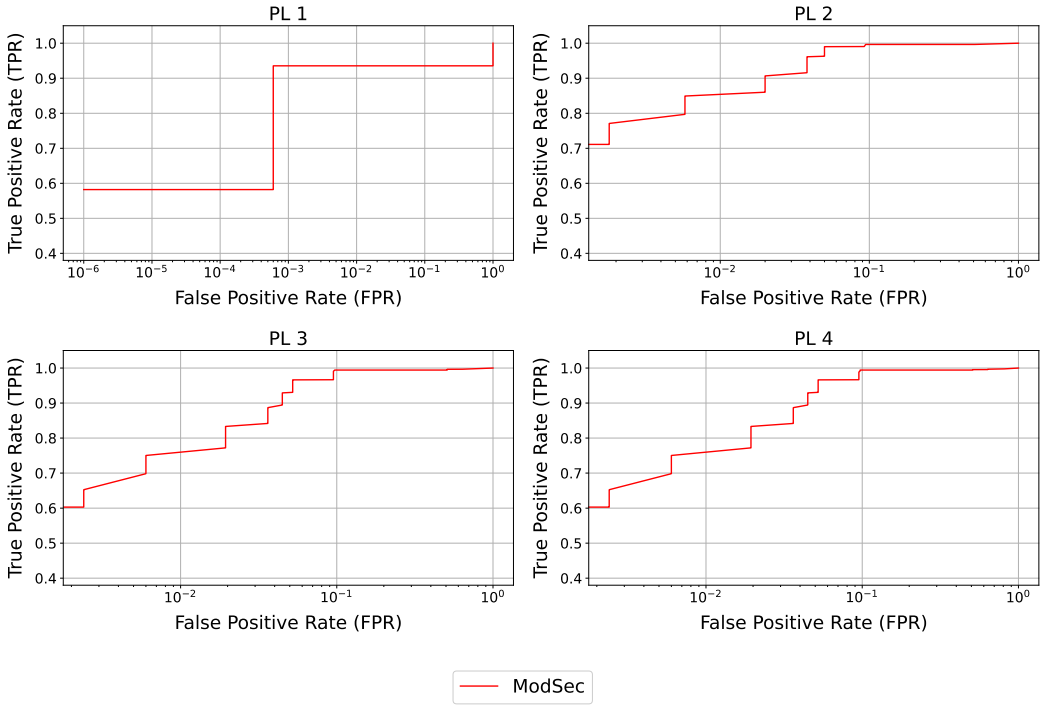
The approach has been evaluated on SQLi attacks but it is general and extends to other attack categories as well



ModSec-Learn: Boosting ModSecurity with Machine Learning - https://arxiv.org/abs/2406.13547

Performance Evaluation

Vanilla ModSecurity



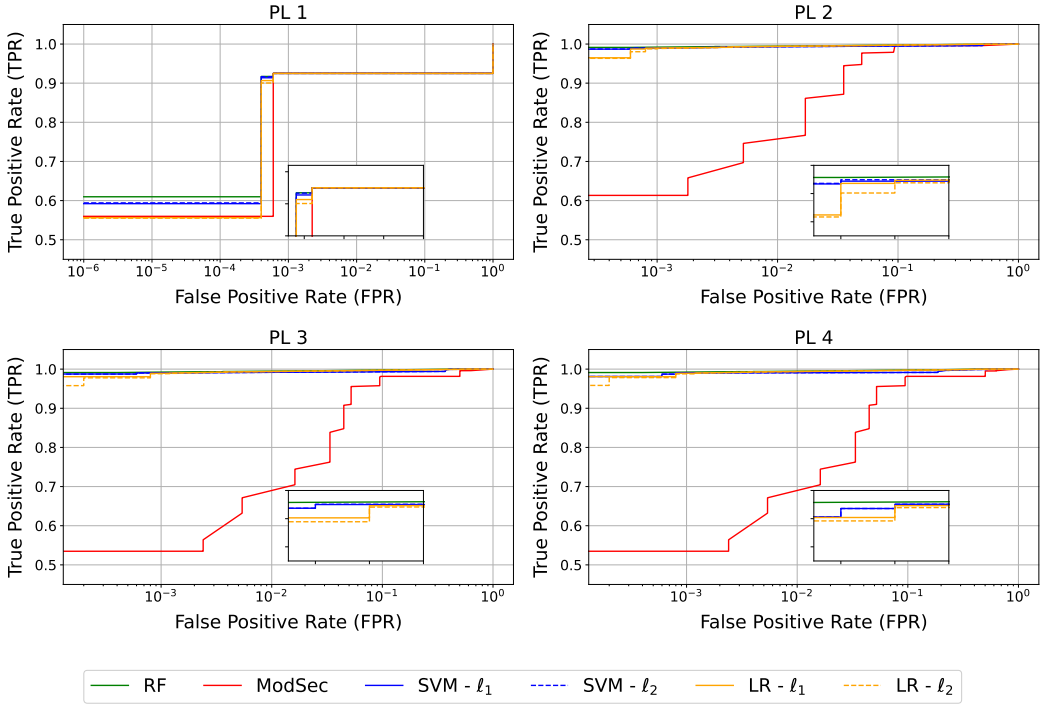
Performance Evaluation

Detection rate @1% False Positives

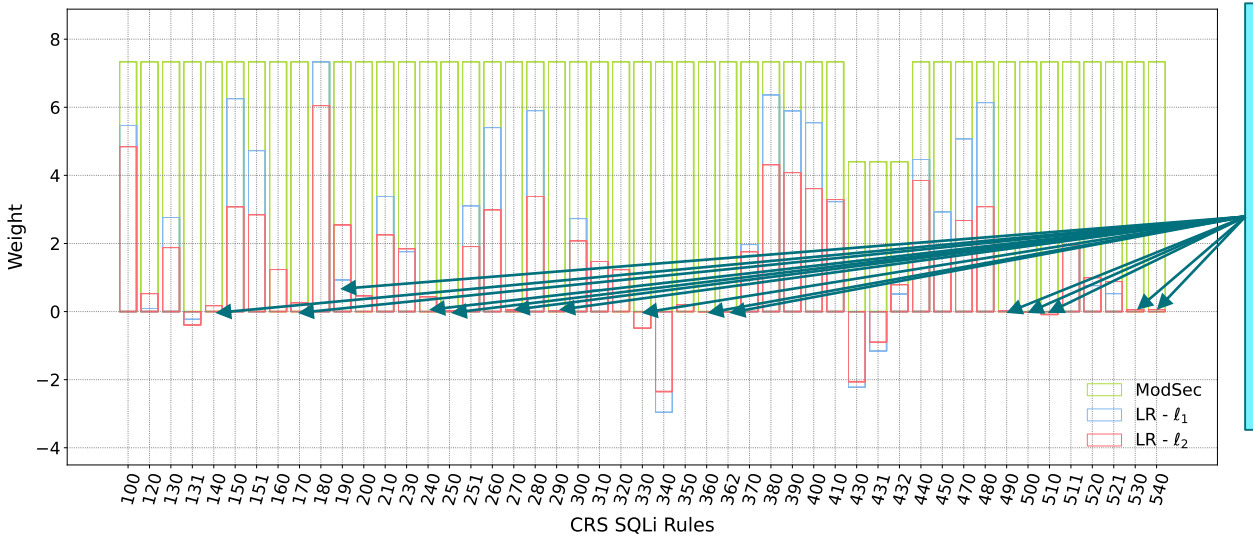
	PL1	PL2	PL3	PL4
ModSec vanilla	**92.50%**	75.45%	68.55%	68.55%
ModSec-Learn SVM (ℓ_1)	92.50%	**99.22%**	99.04%	99.02%
ModSec-Learn SVM (ℓ_2)	92.50%	**99.22%**	99.04%	99.02%
ModSec-Learn LR (ℓ_1)	92.50%	99.34%	99.35%	**99.35%**
ModSec-Learn LR (ℓ_2)	92.50%	99.34%	99.34%	**99.34%**
ModSec-Learn RF	92.50%	99.41%	99.45%	**99.45%**

Performance Evaluation

Lowering the false positive rate



Impact on the rules severity



Weight values learned at PL 4 by ModSec-Learn LR - l1 (blue) and ModSec-Learn LR - l2 (light red), and the weight used by ModSecurity vanilla (green).

- L1 regularization turns off a number of detection rules selecting the most relevant for the detection
- Higher granularity in the severity scores is also suggested by the plot

Original Research Results #2

Making the CRS Robust against Adversarial Attacks (AdvModSec)

Facing Adversarial Attacks

In the context of WAFs, the problem of finding SQLi attacks that are able to bypass the target WAF is *adversarial* in nature → **adversaries manipulate samples to evade detection**

```
SecRule REQUEST_COOKIES|!REQUEST_COOKIES:/__utm/|!REQUEST_COOKIES:/_pk_ref/|REQUEST_COOKIES_NAMES|ARGS_NAMES|ARGS|XML:/* "@rx
      (?:/\*!?|\*/|[';]--|--[\s\r\n\v\f]|--[^-]*?-|[^&-]#.*?[\s\r\n\v\f]|;?\\x00)"
    "id:942440,
    block,
    msg:'SQL Comment Sequence Detected',
    logdata:'Matched Data: %{TX.0} found within %{MATCHED_VAR_NAME}: %{MATCHED_VAR}',
    tag:'attack-sqli ',
    tag:'paranoia-level/2',
    ver:'OWASP_CRS/3.3.4',
    severity:'CRITICAL',
    setvar:'tx.anomaly_score_pl2=+%{tx.critical_anomaly_score}',
    setvar:'tx.sql_injection_score=+%{tx.critical_anomaly_score}'"
```

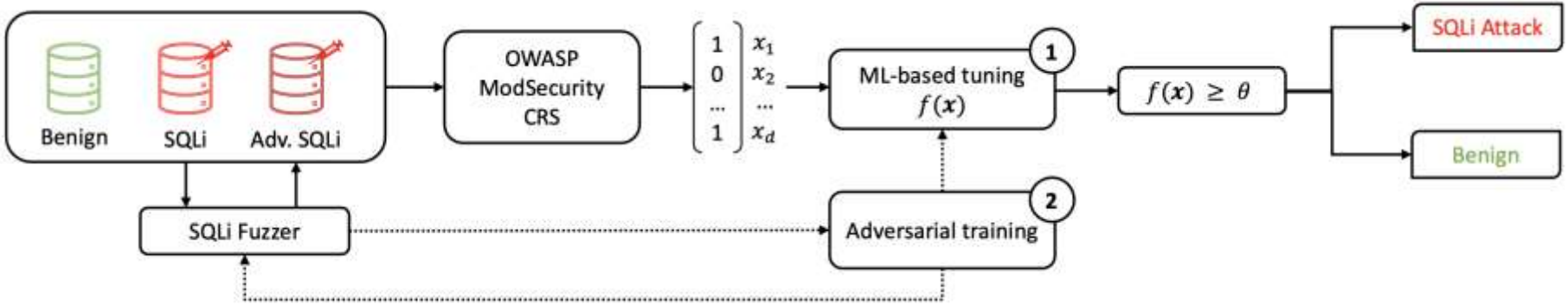
Detected by rule 942440 → 1

NOT Detected by rule 942440 → 2

```
admin' OR 1=1;--'
admin' OR 1=1;  --'
```

Adversarial ModSecurity

Countering Adversarial SQL Injections with Robust Machine Learning

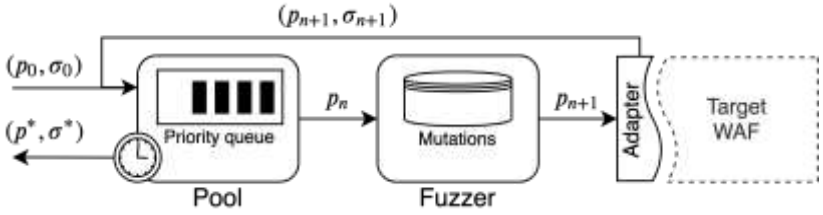


Adversarial ModSecurity: Countering Adversarial SQL Injections with Robust Machine Learning
https://arxiv.org/abs/2308.04964

WAF-a-Mole

Manipulations

Mutation	Example
Case Swapping	`admin' OR 1=1#` ⇒ `admin' oR 1=1#`
Whitespace Substitution	`admin' OR 1=1#` ⇒ `admin'\t\rOR\n1=1#`
Comment Injection	`admin' OR 1=1#` ⇒ `admin'/**/OR 1=1#`
Comment Rewriting	`admin'/**/OR 1=1#` ⇒ `admin'/*xyz*/OR 1=1#abc`
Integer Encoding	`admin' OR 1=1#` ⇒ `admin' OR 0x1=(SELECT 1)#`
Operator Swapping	`admin' OR 1=1#` ⇒ `admin' OR 1 LIKE 1#`
Logical Invariant	`admin' OR 1=1#` ⇒ `admin' OR 1=1 AND 0<1#`



Optimizers

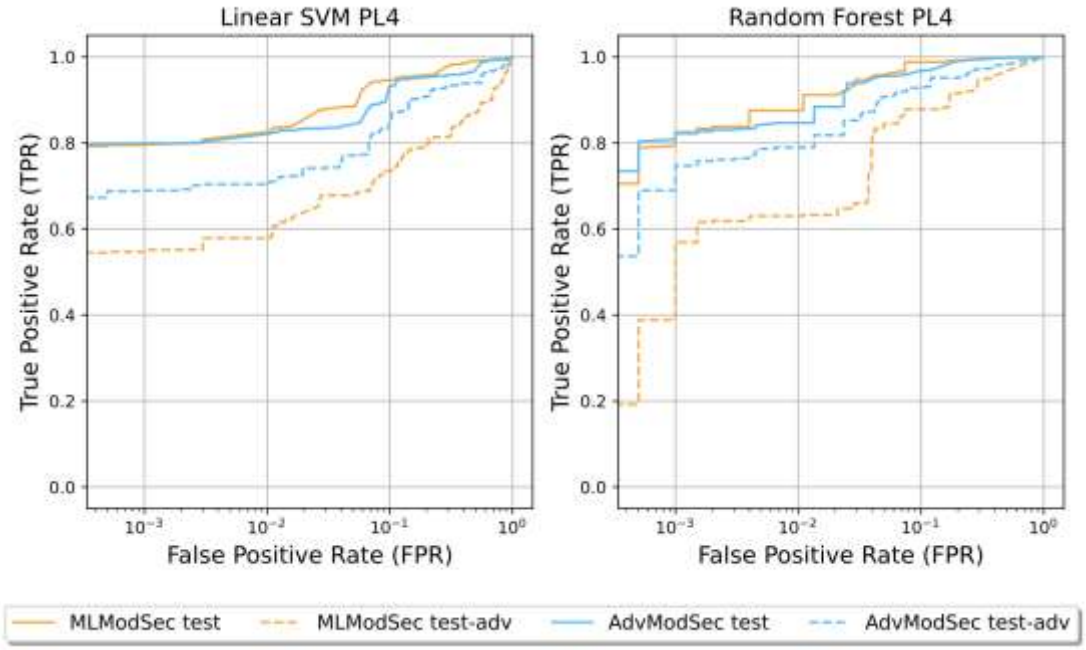
- Guided mutational fuzzer
- Random

L. Demetrio et al. "**WAF-A-MoLE: evading web application firewalls through adversarial machine learning**", 2020

https://github.com/AvalZ/WAF-A-MoLE

Performance Evaluation

Evaluating Robustness against Adversarial Attacks



Final remarks

- Shown two ways to integrate ML in the CRS/ModSecurity decision process
  - **ModSec-Learn** to estimate the severity of the rules based on the traffic
  - **AdvModSec** to make the CRS & ModSecurity resilient against adversarial attacks

- The **ModSec-Learn** approach can be implemented using a simple, linear, classifier
  - No need for integration: just use the *weights* provided by the classifier as *severity* values for the rules
  - Check the code at **https://github.com/pralab/modsec-learn**

More at OWASP AppSec Lisbona 2024

Extended Version of this Talk

https://www.youtube.com/watch?v=LfQBIN6xYQY

New Project - OWASP WARM

WAF Advanced Ruleset Management

Stay tuned on
https://owasp.org/www-project-waf-advanced-ruleset-management/

Q&A

Security Summit

Cagliari, 18 settembre 2024

Contatti

davide.ariu@pluribus-one.it