Trustwave® SpiderLabs®

# From good to excellent

## An overview on the upcoming release of libModSecurity

**Victor Hora (@victorhora) & Felipe Zimmerle (@zimmerle)**

**November 13, 2017**

# Agenda Slide

1. Who are you?
2. WAFs and ModSecurity
3. voilà libModSecurity
4. Info & support
5. Questions


# Victor Hora

**Security Researcher @ Trustwave**

- Member of Spiderlabs Research Team
  - Web Server Security Team
  - Support, development, and consulting for ModSecurity
  - Contributor to the OWASP Core Rule Set

- Prior
  - Security Consultant / Penetration Tester (PS&C Group Australia)
  - Security Engineer at Nagra (Threat Intelligence / Forensics for SetTop Boxes)
  - Security Consultant at Tempest Security (where I met Felipe 12 years ago! ☺)

# Felipe Zimmerle

**Lead Developer / Security Researcher @ Trustwave**

- Member of Spiderlabs Research Team
  - Architect and lead developer for ModSecurity
  - Lead researches in WebApp Security
  - Contributor to the OWASP Core Rule Set

- Other
  - Current: Ph.D student in Blockchains and WAFs
  - Prior :
    - R&D 3D Printing and GSM Security, GrSec/SELinux (ARM)
    - MaeMO / MeeGo Software Engineer for Nokia
    - Security Consultant at Tempest Security (where I met Victor 12 years ago! ☺)

# WAFs and ModSecurity

# WAF and it's many industry names

Adaptive Firewall

Adaptive Gateway

Application Firewall

Application-layer Firewall

Application-level Security Gateway

Application Level Gateway

Application Security Device

Application Security Gateway

Stateful Multilayer Inspection Firewall

Web Adaptive Firewall

Web Application Controller

Web Application Firewall

Web Application Security Device

Web Application Proxy

Web Application Shield

Web Security Firewall

Web Security Gateway

Web Security Proxy

Web Intrusion Detection System

Web Intrusion Prevention System

**WAF:** can be a virtual or physical appliance that prevents vulnerabilities in web applications from being exploited by outside threats

”By 2020, more than 60 percent of public web applications will be protected by a WAF”

Gartner

# WAF Features

- Protocol Anomaly Detection
- Enforcing Input Validation
- Negative vs. Positive Security Models
- Rule-Based vs. Anomaly-Based Models
- State Management Protections
- Response Monitoring/Information Leak Protection
- Virtual Patching

# **ModSecurity:** The Swiss Army Knife of WAFs.

# Representative users of ModSecurity?


Westpac

AT&T

UNIVERSITY OF OXFORD

intel

YOTTAA

CISCO

TEXAS INSTRUMENTS

amadeus

Fender

cPanel

GENERAL DYNAMICS

section.io

Caltech

Microsoft

BLUE COAT

DELL

KEMP

vmware

NGINX

Akamai

New Zealand Post

NEW YORK UNIVERSITY

TOSHIBA

COMODO

verizon

edgecast

ERICSSON

fastly

The Walt Disney Company

BOEING

Palantir

CLOUDFLARE

IMPERVA INCAPSULA

CGI

Bell

# ModSecurity features

**An Open Source Multiplatform Web Application Firewall**

- Version 2 currently supports Apache, Nginx and IIS
- ~~It's a web filter module that relies on Apache~~

**Most widely deployed WAF:**

- Over 10,000 installations (2011)
- Many vendors and third parties uses ModSecurity as engine

**Deep understanding and auditing of (HTTP/HTML):**

- Encodings / Multipart
- XML
- JSON
- Full audit HTTP traffic capability (full request / response )

**Ideal for Virtual Patching:**

- Ability to enforce both Negative and Positive Security Models
- Highly accurate rules mitigating known vulnerabilities

modsecurity

Open Source Web Application Firewall

# Open Source

### Agility

- New features and rules can be quickly developed and released to the public for use.

### Availability

- Since there is no **up front** cost, anyone can download and use it right away.

### Collaborative Development

- Can leverage community resources for developing new features. (Apache License)

### Transparency

- Since the code is readily available, users can be sure what it does.

### Education

- Allows for web application security presentations at security conferences (ahem) that don't want "Vendor Pitches" ☺

# What it doesn't offer

### ModSecurity doesn't come with a ruleset

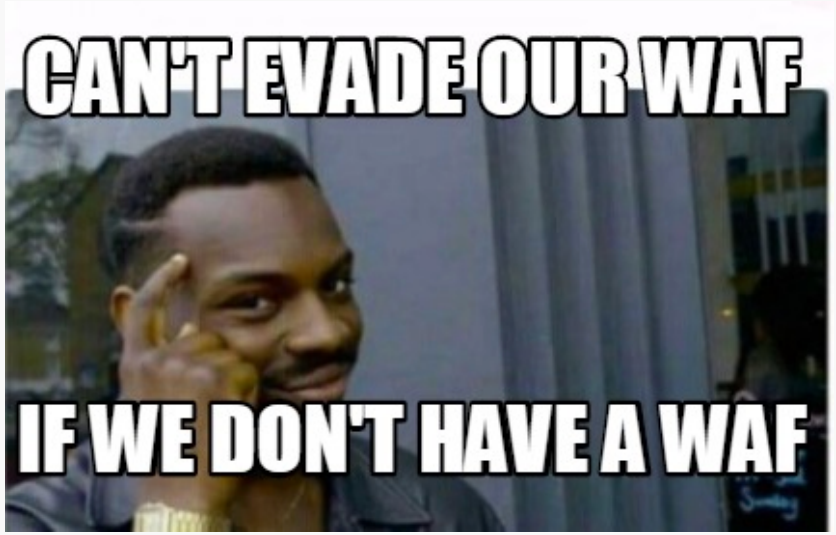
- That's right by default it stops almost nothing
- OWASP Core Rule Set Project
- Commercial Rules

### No GUI

- Logging is available and configurable in many forms
- If you want to visualize your output you need to output results to some third-party tool (e.g. WAF FLE)

### No rule development platform

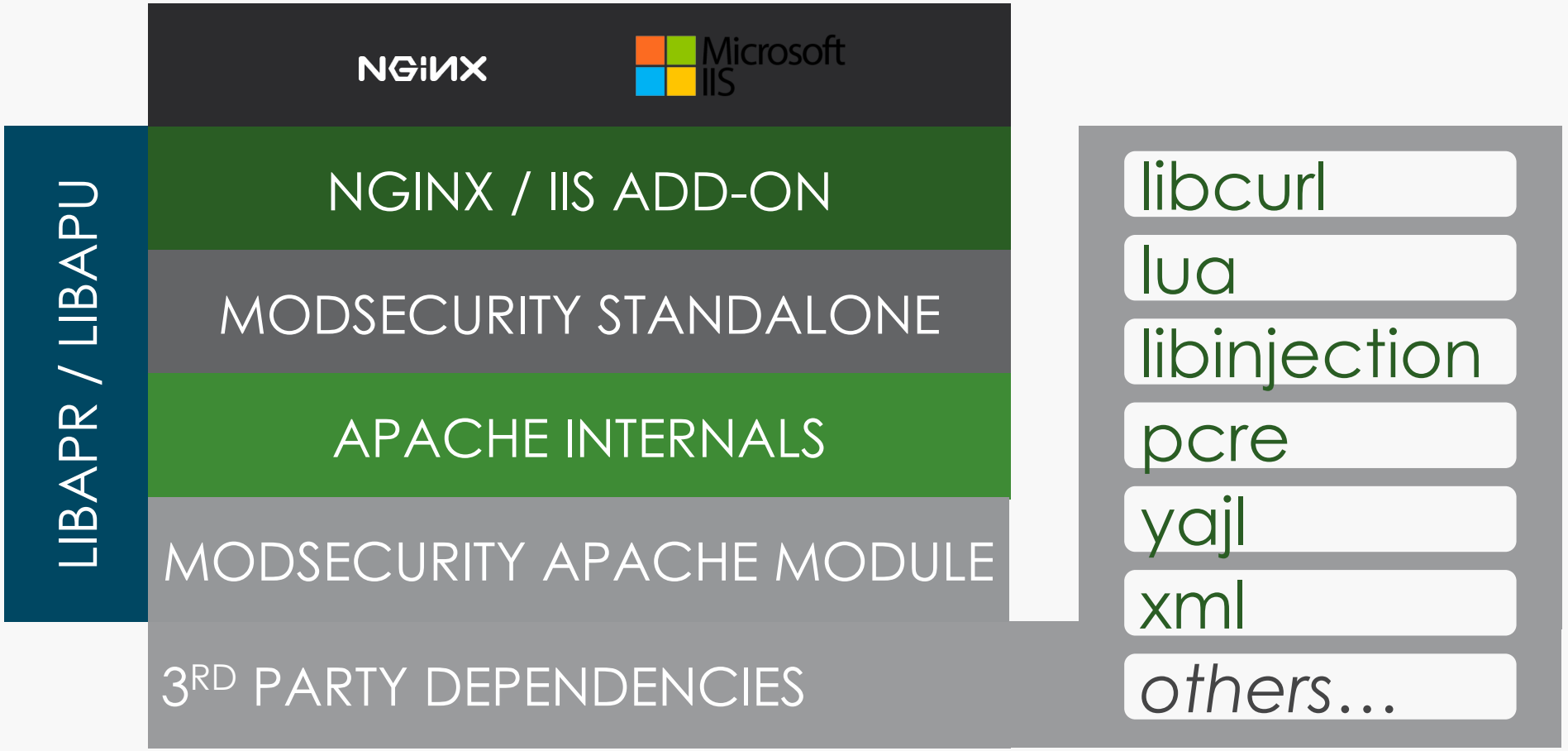
- As shipped any rules you develop are done by hand
- SecRemoteRules



# Ok cool. So why rewrite ModSecurity from scratch!?


modsecurity
Open Source Web Application Firewall

# ModSecurity v2.9 architecture



# ModSecurity for IIS

**Refactoring of IIS build scripts**

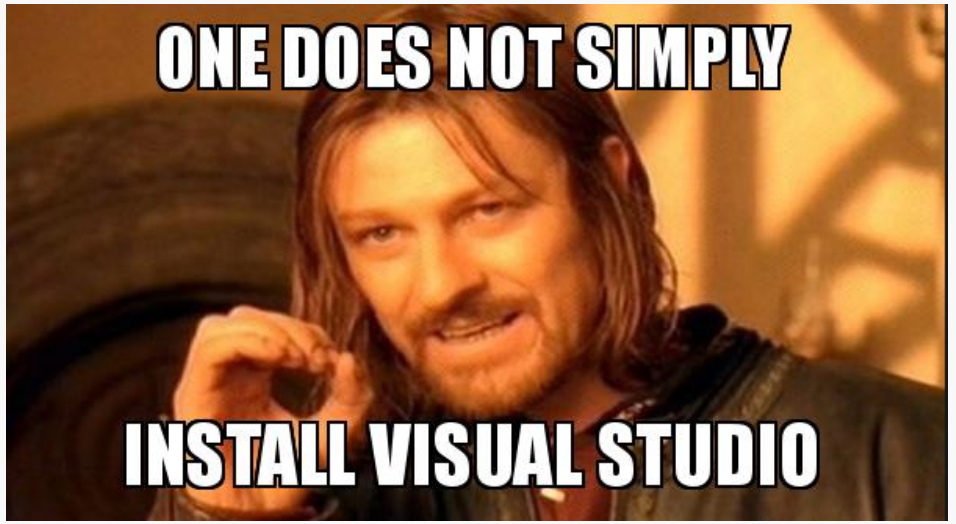
v2/master v2.9.2

victorhora committed with zimmerle on Jul 17 Verified 1 parent 61bce8d

Showing 7 changed files with 181 additions and 86 deletions.

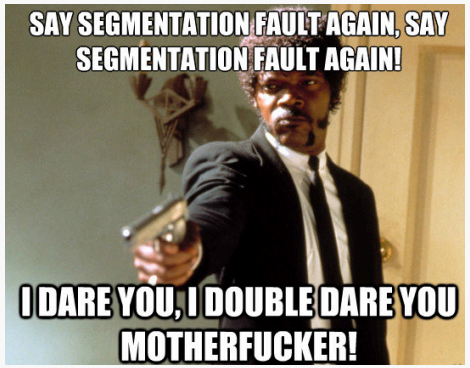
43 iis/build_dependencies.bat

```
@@ -1,25 +1,30 @@
-:: Those variable should be edited as needed.
+:: Those variables should be edited as needed.
 :: Use full paths.

 :: General paths
 @set WORK_DIR=%cd%\dependencies\build_dir
 @set OUTPUT_DIR=%cd%\dependencies\release_files
 @set SOURCE_DIR=%USERPROFILE%\Downloads

+:: Dependencies
+@set CMAKE=cmake-3.8.2-win32-x86.zip
+@set PCRE=pcre-8.40.zip
+@set ZLIB=zlib-1.2.11.tar.gz
+@set LIBXML2=libxml2-2.9.4.tar.gz
+@set LUA=lua-5.3.4.tar.gz
+@set CURL=curl-7.54.1.zip
+@set APACHE_SRC=httpd-2.4.27.tar.gz
+@set APACHE_BIN32=httpd-2.4.27-win32-VC11.zip
+@set APACHE_BIN64=httpd-2.4.27-win64-VC11.zip
+@set YAJL=yajl-2.1.0.zip
+@set SSDEEP=ssdeep-2.13.tar.gz
+@set SSDEEP_BIN=ssdeep-2.13.zip
+
+@set CMAKE_DIR=%WORK_DIR%\%CMAKE:~0,-4%\bin
```



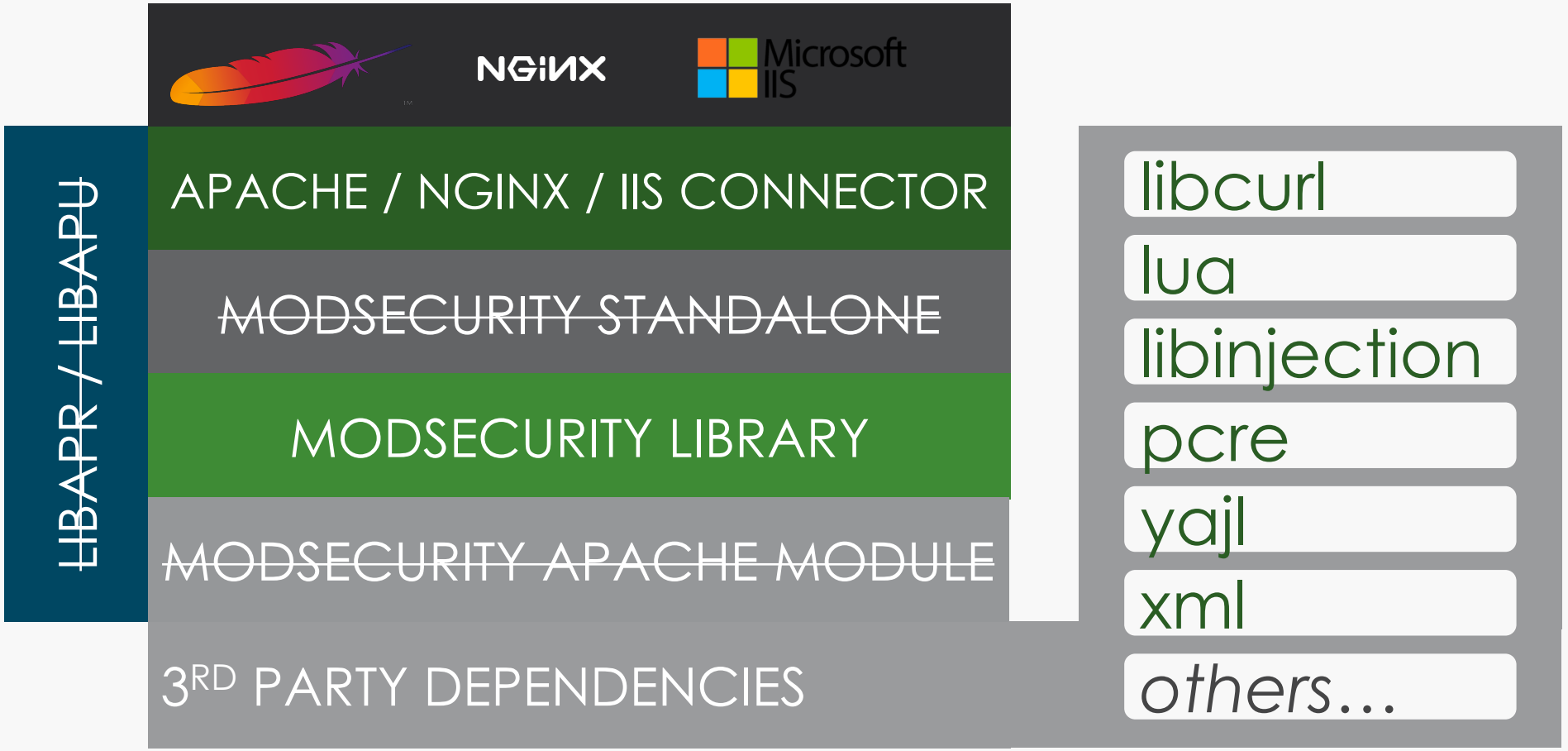
# ModSecurity for Nginx

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 6 Open ✓ 58 Closed | Author | Labels | Projects | Milestones | Assignee | Sort |
| | Nginx with Modsecurity: POST request gives 500 error Bug Platform - Nginx TBF by libmodsec<br>#582 by pijnack was closed on May 9 | | | | | | 141 |
| | POST body is not inspected by modsecurity TBF by libmodsec<br>#684 by code1955 was closed on May 9 | | | | | | 41 |
| | file uploads over 8k fail when using ModSecurity 2.7.5 and Nginx 1.4.2 Bug Platform - Nginx TBF by libmodsec<br>#142 by wellumies was closed on May 9 | | | | | | 37 |
| | NGINX and SecRequestBodyAccess On option, don't pass POST request to Upstream Server Bug Platform - Nginx TBF by libmodsec<br>#664 by zotgene was closed on May 9 | | | | | | 34 |
| | modsecurity nginx coredump TBF by libmodsec<br>#839 by shel3over was closed on May 9 | | | | | | 30 |
| | Compiling for NGINX without Apache TBF by libmodsec<br>#686 by infinitnet was closed on May 9 | | | | | | 24 |
| | Nginx 1.5.10 + ModSecurity 2.7.7: Process killed by segmentation fault at 10 Platform - Nginx TBF by libmodsec<br>#658 by GinoHerelam was closed on May 9 | | | | | | 18 |
| | Nginx mod_security leaks file descriptors Bug libmodsecurity Platform - Nginx TBF by libmodsec<br>#137 by kirilkalchev was closed on May 5 | | | | | | 16 |
| | nginx: intentionally removed response headers are added back in TBF by libmodsec<br>#853 by eyz was closed on May 9 | | | | | | 13 |
| | Disabling mlogc breaks standalone build TBF by libmodsec<br>#900 by quanah was closed on May 4 | | | | | | 13 |



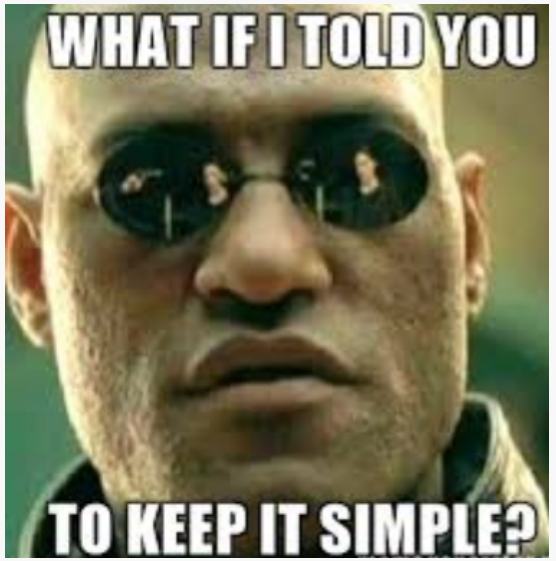
# Voilà libModSecurity

# libModSecurity (aka 3.0) architecture



# libModSecurity install: simple

```
# apt-get install g++ flex bison curl doxygen
libyajl-dev libgeoip-dev libtool dh-autoreconf
libcurl4-gnutls-dev libxml2 libpcre++-dev libxml2-
dev

$ sh build.sh

$ git submodule init && git submodule update

$ ./configure

$ make

$ make install
```



# A simple libModSecurity connector

```
#include "modsecurity/modsecurity.h"
#include "modsecurity/transaction.h"

char main_rule_uri[] = "basic_rules.conf";

int main (int argc, char **argv)
{
    ModSecurity *modsec = NULL;
    Transaction *transaction = NULL;
    Rules *rules = NULL;

    modsec = msc_init();

    rules = msc_create_rules_set();
    msc_rules_add_file(rules, main_rule_uri);

    transaction = msc_new_transaction(modsec, rules, NULL);

    msc_process_connection(transaction, "127.0.0.1", 12345, "127.0.0.1", 80);
    msc_process_uri(transaction, "http://www.modsecurity.org/test?key1=value1&test=test", "GET", "1.1");
    msc_process_request_headers(transaction);
    msc_process_request_body(transaction);
    msc_process_response_headers(transaction, 200, "HTTP 1.3");
    msc_process_response_body(transaction);

    return 0;
}
~
```

SecRule "ARGS @detectXSS" "id:10,pass"

# A simple libModSecurity connector

```
root@ubuntu:/root/ModSecurity-v3-latest/examples/simple_example_using_c# ./test ; cat debug.log
Server log callback is not set -- [client 127.0.0.1] ModSecurity: Warning. detected XSS using l
ibinjection. [file "basic_rules.conf"] [line "10"] [id "10"] [rev ""] [msg ""] [data ""] [sever
ity "0"] [ver ""] [maturity "0"] [accuracy "0"] [ref "v53,16"] [hostname "127.0.0.1"] [uri "htt
p://www.modsecurity.org/test"] [unique_id "151035796141.374546"]
[9] JSON parser initialization
[9] yajl JSON parsing callback initialization
[4] Initializing transaction
[4] Transaction context created.
[4] Starting phase CONNECTION. (SecRules 0)
[9] This phase consists of 0 rule(s).
[4] Starting phase URI. (SecRules 0 + 1/2)
[4] Adding request argument (GET): name "key1", value "value1"
[4] Adding request argument (GET): name "test", value "<script>alert(1)"
[4] Starting phase REQUEST_HEADERS.  (SecRules 1)
[9] This phase consists of 1 rule(s).
[4] (Rule: 10) Executing operator "DetectXSS" with param "" against ARGS.
[9] Target value: "value1" (Variable: ARGS:key1)
[9] libinjection was not able to find any XSS in: value1
[9] Target value: "<script>alert(1)" (Variable: ARGS:test)
[5] detected XSS using libinjection.
[9] Matched vars updated.
[9] Rule contains a `pass' action
[4] Rule returned 1.
[4] Not running disruptive action: pass. SecRuleEngine is not On
[4] Starting phase REQUEST_BODY. (SecRules 2)
[4] Request body processing is disabled
[9] This phase consists of 0 rule(s).
[4] Starting phase RESPONSE_HEADERS. (SecRules 3)
[9] This phase consists of 0 rule(s).
[4] Starting phase RESPONSE_BODY. (SecRules 4)
[9] This phase consists of 0 rule(s).
```

# DEMO

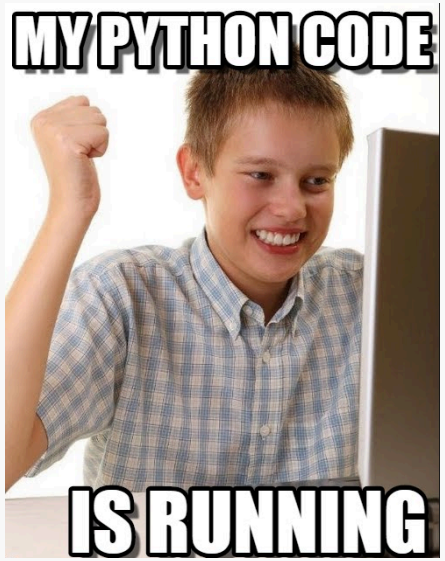
# Can you make it simpler to extend?

# Python bindings for libModSecurity

```
m modsecurity import *

modsec = ModSecurity()
rules = Rules()
r = rules.loadFromUri("/path/to/your/v3.0.0-dev/rules/REQUEST-10-IP-REPUTATION.conf")
if r == -1:
    print rules.getParserError()
    sys.exit()

i = 0
while i < modsec.NUMBER_OF_PHASES:
    r =  rules.getRulesForPhase(i)
    print "-- Phase " + str(i)
    for x in r:
        if x.rule_id == 0:
            continue

        print "    Rule Id: " + str(x.rule_id)
        print "       From: " + str(x.m_fileName) + " at " + str(x.m_lineNumber)
    i = i + 1
~
```

# Python bindings for libModSecurity

```
-- Phase 1
-- Phase 2
-- Phase 3
 Rule Id: 981140
    From: /tmp/owasp-modsecurity-crs/rules/REQUEST-10-IP-REPUTATION.conf at 1
 Rule Id: 900050
    From: /tmp/owasp-modsecurity-crs/rules/REQUEST-10-IP-REPUTATION.conf at 36
 Rule Id: 900051
    From: /tmp/owasp-modsecurity-crs/rules/REQUEST-10-IP-REPUTATION.conf at 68
 Rule Id: 981138
    From: /tmp/owasp-modsecurity-crs/rules/REQUEST-10-IP-REPUTATION.conf at 115
 Rule Id: 981141
    From: /tmp/owasp-modsecurity-crs/rules/REQUEST-10-IP-REPUTATION.conf at 141
 Rule Id: 981142
    From: /tmp/owasp-modsecurity-crs/rules/REQUEST-10-IP-REPUTATION.conf at 167
 Rule Id: 981143
    From: /tmp/owasp-modsecurity-crs/rules/REQUEST-10-IP-REPUTATION.conf at 192
 Rule Id: 981144
    From: /tmp/owasp-modsecurity-crs/rules/REQUEST-10-IP-REPUTATION.conf at 217
 Rule Id: 981139
    From: /tmp/owasp-modsecurity-crs/rules/REQUEST-10-IP-REPUTATION.conf at 242
-- Phase 4
-- Phase 5
-- Phase 6
```



# Lua scripting engine

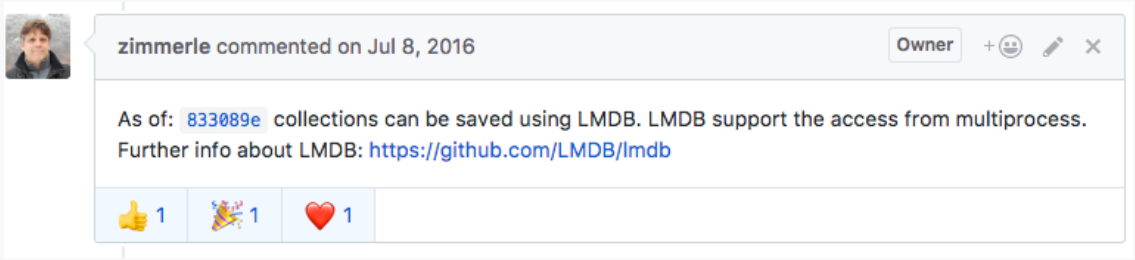
SecRuleScript "script.lua" "id:11,deny"

**Script.lua**

```
function main()
local d = m.getvars("ARGS", { "lowercase", "htmlEntityDecode" } );

for i = 1, #d do
if (string.find(d[i].value, "<script"))
return ("Suspected XSS in variable " .. d[i].name .. ".");
      end
return nil;
 end
```

# Can you make it faster?

# libModSecurity collections backend




- Key/Value store using B+trees
- Fully transactional, ACID compliant
- MVCC, readers nerver block
- Users memory-mapped files, needs no tuning
- Crash-proof, no recovery needed after restart
- Highly optimized, extremely compact
  - <40KB obj code, fits in CPU L1


# Collections backend – more to come

Implement Redis support as Collection backend on libmodsecurity #1139

Open **zimmerle** opened this issue on May 4, 2016 · 4 comments

**zimmerle** commented on May 4, 2016 — Owner

ModSecurity version 3 architecture allow the utilization of multiple backends, including redis. The support should be implemented. The interface is available here:

https://github.com/SpiderLabs/ModSecurity/blob/libmodsecurity/headers/modsecurity/collection/collection.h

4 2

Implement Memcache support as Collection backend on libmodsecurity #1140

Open **zimmerle** opened this issue on May 4, 2016 · 2 comments

**zimmerle** commented on May 4, 2016 • edited — Owner

ModSecurity version 3 architecture allow the utilization of multiple backends, including memcache. The support should be implemented. The interface is available here:

https://github.com/SpiderLabs/ModSecurity/blob/libmodsecurity/headers/modsecurity/collection/collection.h

1

redis

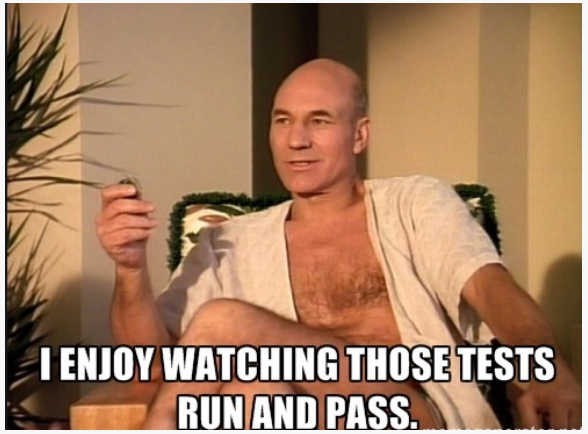
memCached

# DEMO

# Can you make it safer?

# libModSecurity unit tests:

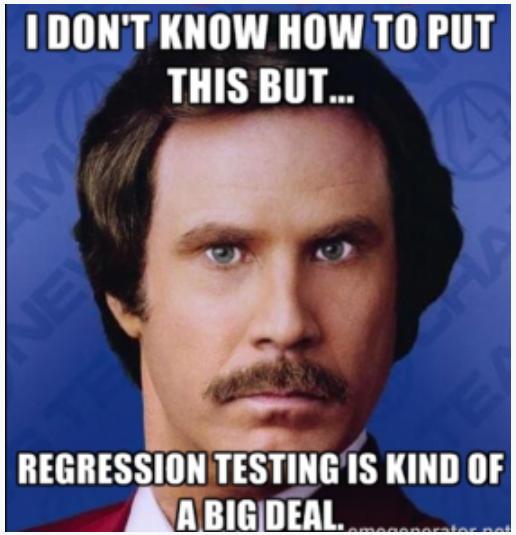
```
  test-cases/secrules-language-tests/transformations/utf8toUnicode.json:utf8toUnicode...        0 tests failed.
Total >> 4278

Ran a total of: 4278 unit tests - All tests passed
==66039==
==66039== HEAP SUMMARY:
==66039==     in use at exit: 72,704 bytes in 1 blocks
==66039==   total heap usage: 330,235 allocs, 330,234 frees, 150,759,768 bytes allocated
==66039==
==66039== LEAK SUMMARY:
==66039==    definitely lost: 0 bytes in 0 blocks
==66039==    indirectly lost: 0 bytes in 0 blocks
==66039==      possibly lost: 0 bytes in 0 blocks
==66039==    still reachable: 72,704 bytes in 1 blocks
==66039==         suppressed: 0 bytes in 0 blocks
==66039== Reachable blocks (those to which a pointer was found) are not shown.
==66039== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==66039==
==66039== For counts of detected and suppressed errors, rerun with: -v
==66039== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
root@ubuntu:/root/ModSecurity-v3-latest/test#
```



# libModSecurity regression testing

```
393 variable-UNIQUE_ID.json                    Testing Variables :: UNIQUE_ID                          passed!
394 variable-URLENCODED_ERROR.json             Testing Variables :: URLENCODED_ERROR - GET (1/7)       passed!
395 variable-URLENCODED_ERROR.json             Testing Variables :: URLENCODED_ERROR - GET (2/7)       passed!
396 variable-URLENCODED_ERROR.json             Testing Variables :: URLENCODED_ERROR - GET (6/7)       passed!
397 variable-URLENCODED_ERROR.json             Testing Variables :: URLENCODED_ERROR - GET (7/7)       passed!
398 variable-URLENCODED_ERROR.json             Testing Variables :: URLENCODED_ERROR - POST (3/7)      passed!
399 variable-URLENCODED_ERROR.json             Testing Variables :: URLENCODED_ERROR - POST (4/7)      passed!
400 variable-URLENCODED_ERROR.json             Testing Variables :: URLENCODED_ERROR - POST (5/7)      passed!
401 variable-USERID.json                       Testing USERID variable (1/2)                           passed!
402 variable-USERID.json                       Testing USERID variable (2/2)                           passed!
403 variable-WEBSERVER_ERROR_LOG.json          Testing Variables :: WEBSERVER_ERROR_LOG (1/1)          passed!
404 variable-variation-count.json              Testing variable variations :: count (1/3)              passed!
405 variable-variation-count.json              Testing variable variations :: count (2/3)              passed!
406 variable-variation-count.json              Testing variable variations :: count (3/3)              passed!
407 variable-variation-exclusion.json          Testing variable variations :: exclusion (1/2)          passed!
408 variable-variation-exclusion.json          Testing variable variations :: exclusion (2/2)          passed!

Ran a total of: 408 regression tests - All tests passed. 0 skipped test(s). 0 disabled test(s).
root@ubuntu:/root/ModSecurity-v3-latest/test#
```

```
(3623/  0/3623): test/test-cases/secrules-language-tests/operators/ipMatch.json
(   5/  0/   5): test/test-cases/secrules-language-tests/operators/strmatch.json
(   3/  0/   3): test/test-cases/secrules-language-tests/operators/detectXSS.json
(  13/  0/  13): test/test-cases/secrules-language-tests/operators/eq.json
(  49/  0/  49): test/test-cases/regression/offset-variable.json
(   1/  0/   1): test/test-cases/regression/config-update-target-by-tag.json
(   1/  0/   1): test/test-cases/regression/config-update-target-by-id.json
================================================================================
Testsuite summary for modsecurity 3.0
================================================================================
# TOTAL: 4680
# PASS:  4680
# SKIP:  0
# XFAIL: 0
# FAIL:  0
# XPASS: 0
# ERROR: 0
================================================================================
make[3]: Leaving directory '/root/ModSecurity-v3-latest'
make[2]: Leaving directory '/root/ModSecurity-v3-latest'
make[1]: Leaving directory '/root/ModSecurity-v3-latest'
root@ubuntu:/root/ModSecurity-v3-latest#
```



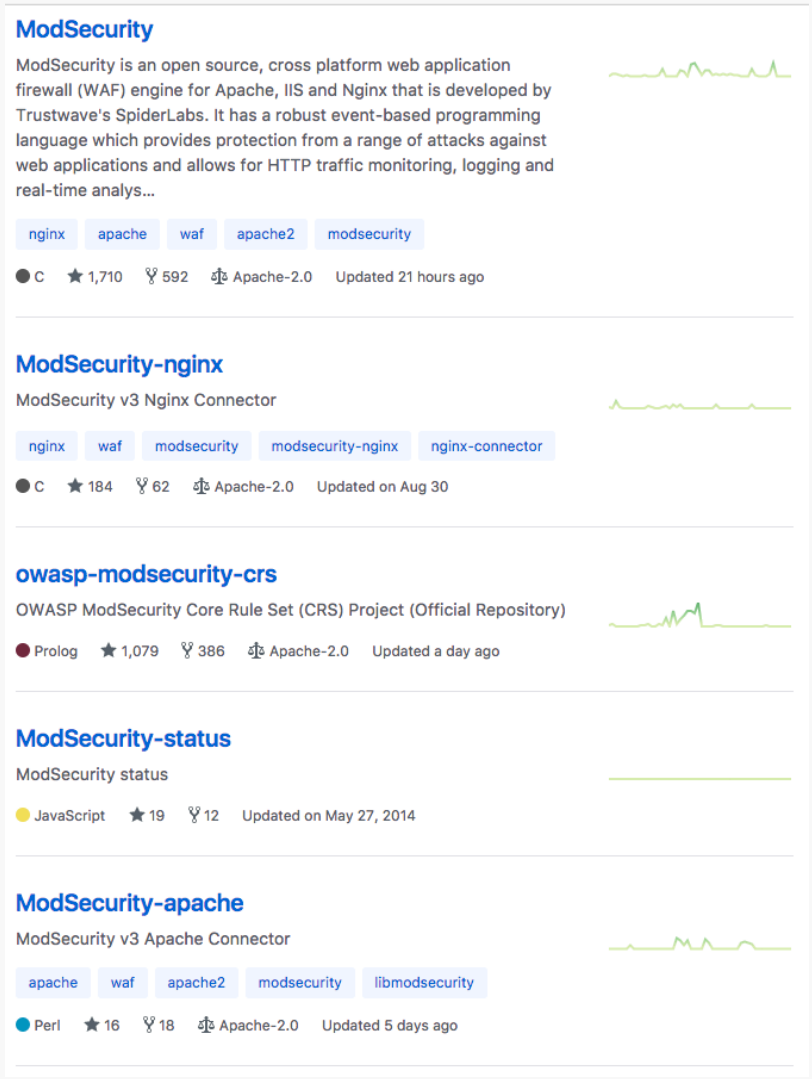
# libModSecurity fuzzing tests

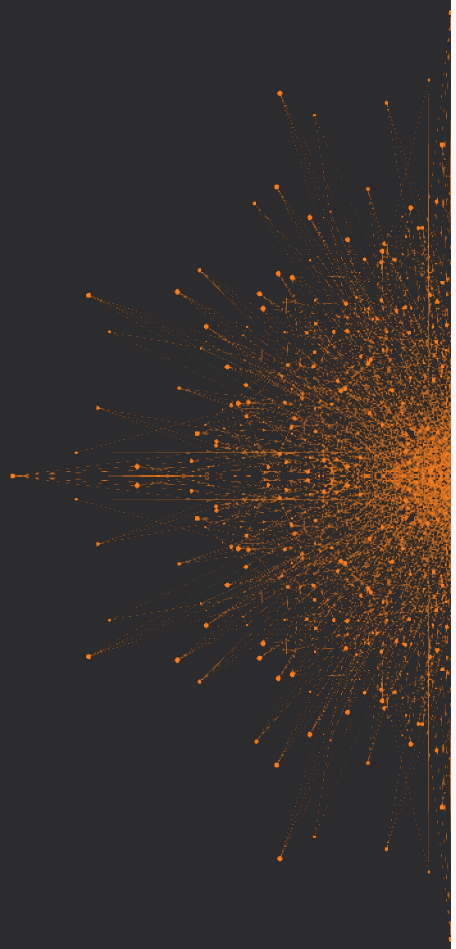
```
                american fuzzy lop 2.41b (fuzzer2)

┌─ process timing ─────────────────────────────────┬─ overall results ─┐
│        run time : 140 days, 19 hrs, 9 min, 1 sec │  cycles done : 1.96M │
│   last new path : 79 days, 13 hrs, 20 min, 40 sec│  total paths : 30    │
│ last uniq crash : none seen yet                  │ uniq crashes : 0     │
│  last uniq hang : none seen yet                  │   uniq hangs : 0     │
├─ cycle progress ─────────────────┬─ map coverage ─┴──────────────────────┤
│  now processing : 29* (96.67%)   │    map density : 3.80% / 4.65%       │
│ paths timed out : 0 (0.00%)      │ count coverage : 1.31 bits/tuple     │
├─ stage progress ─────────────────┼─ findings in depth ──────────────────┤
│  now trying : splice 10          │ favored paths : 9 (30.00%)           │
│ stage execs : 95/128 (74.22%)    │  new edges on : 14 (46.67%)          │
│ total execs : 11.9G              │ total crashes : 0 (0 unique)         │
│  exec speed : 965.6/sec          │  total tmouts : 3.12M (27 unique)    │
├─ fuzzing strategy yields ────────┴──────────────┬─ path geometry ──────┤
│   bit flips : 0/6608, 0/6578, 0/6518           │    levels : 3        │
│  byte flips : 0/826, 0/796, 0/746              │   pending : 0        │
│ arithmetics : 0/46.1k, 0/7688, 0/1694          │  pend fav : 0        │
│  known ints : 0/4522, 0/20.8k, 0/32.3k         │ own finds : 29       │
│  dictionary : 0/0, 0/0, 0/4257                 │  imported : 0        │
│       havoc : 28/5.08G, 1/6.80G                │ stability : 88.26%   │
│        trim : 0.00%/245, 0.00%                 ├──────────────────────┘
└────────────────────────────────────────────────┘             [cpu:393%]
```

# Info & support

# Info

- https://github.com/SpiderLabs/ModSecurity
- https://github.com/SpiderLabs/ModSecurity-nginx
- https://github.com/SpiderLabs/ModSecurity-apache
- https://github.com/SpiderLabs/secrules-language-tests
- https://www.trustwave.com/Resources/SpiderLabs-Blog/?tag=ModSecurity
- www.modsecurity.org
- @ModSecurity
- Modsecurity.slack.com
- #modsecurity (@freenode.net)





# Come work with us ☺!

**Security Researcher - ModSecurity:** To join our SpiderLabs ModSecurity Research Team to supports ModSec and Trustwave WAF.

This position will split time between supporting ModSecurity commercial customers and researching web application threats and countermeasures.

http://app.jobvite.com/m?3vsx8jwq

**Security Researcher -** Vulnerability Assessment Team (Linux)

**Security Researcher -** Database Security Team (DST)

**Security Information Specialist -** SIEM & MSS

# Questions?

Trustwave®

Smart security on demand