BREACH

ModSecurity

The Open Source Web Application Firewall

Ivan Ristic

Chief Evangelist

Breach Security

Introduction

Breach Security

- Global headquarters in Carlsbad, California
- Web application security provider for over six years
- Led by experienced security executives
- Trusted by large enterprise customers

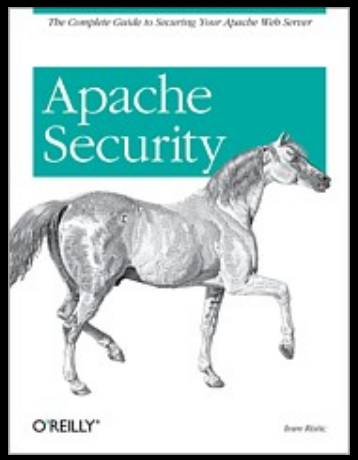
- Next-generation web application security solutions for protecting business-critical applications transmitting privileged information.
- Resolve security challenges such as identity theft, information leakage, regulatory compliance, and insecurely coded applications.
- Best threat detection in the industry and the most flexible deployment options available.

BREACH

Introduction

Ivan Ristic

- **Web application security** and **web application firewall** specialist.
- Author of **Apache Security**.
- Author of **ModSecurity**.
- **OWASP London Chapter** leader.
- Officer of the **Web Application Security Consortium**.
  - WAFEC project leader.



modsecurity

BREACH

BREACH™

Part 1

What are Web Application Firewalls?

Problems with Web Applications

How did it all start?

- HTTP and browsers designed for document exchange.
- Web applications built using a number of loosely integrated technologies.
- No one thought about security at the time.

Where are we today?

- Most web applications suffer from one type of problem or another. It is very difficult to develop a reasonably secure web application.
- Not possible to achieve 100% security.

BREACH

How Can We Improve the Situation?

Education & good development practices.

- We have been working hard on this since 2000.
- Much better than it used to be, but still not good enough.
- Secure web programming too difficult and time consuming for your average programmer.

Design & code reviews.

- Slow and expensive.

Scanning & penetration testing.

- Not conclusive.
- Slow and expensive.

BREACH

Why Use Web Application Firewalls?

It's a cost-effective technology that **works**.

It can be deployed **straight away**.

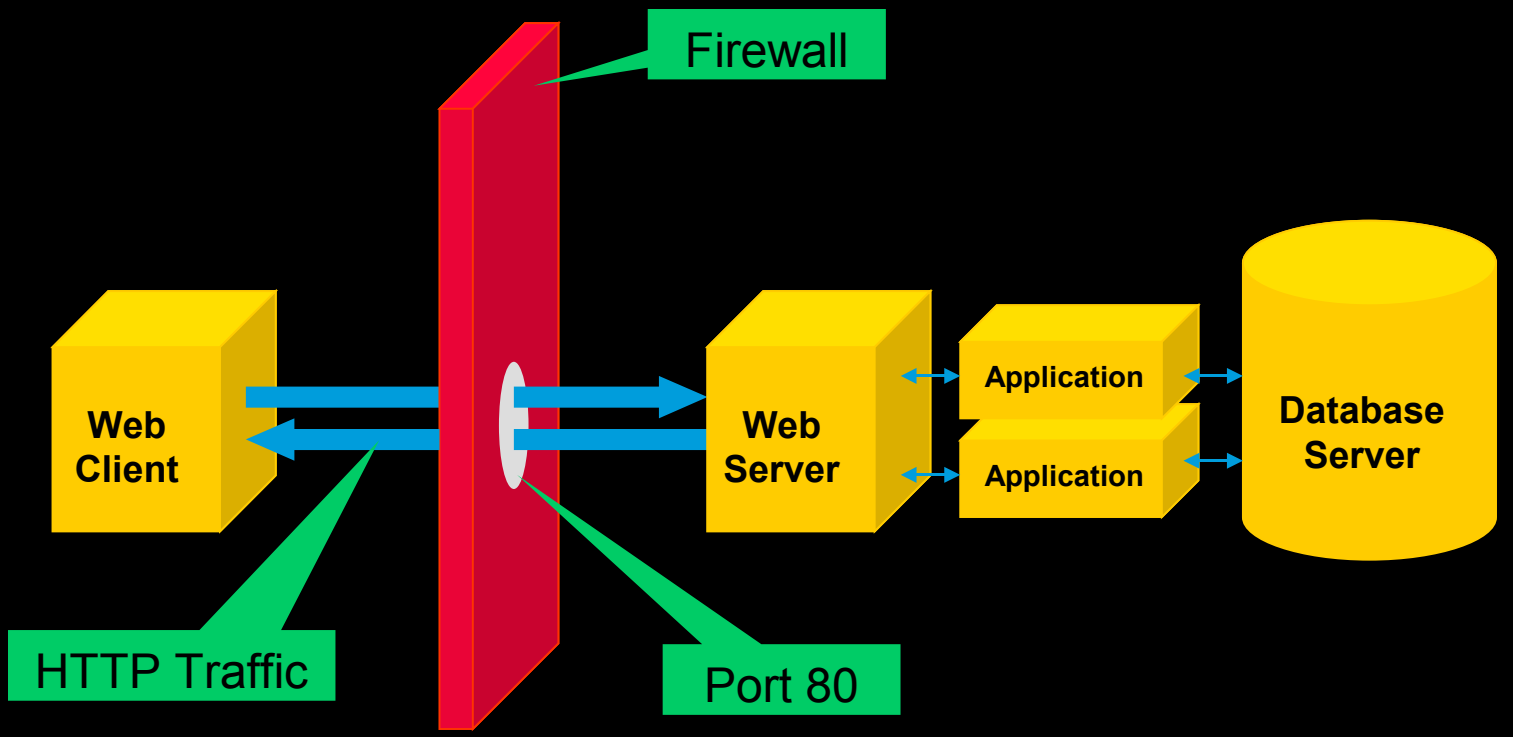
Gives **instant visibility** of the systems it protects.

Can provide instant **protection**.

In some of its forms (reverse proxies) it is actually an **essential building block of HTTP networks**.

Good example of **defence-in-depth**.

Network Firewalls Do Not Work



Neither do IDS/IPS solutions.

BREACH

WAF Identity Problem: Naming

There is a long-standing WAF identity problem.

With the **name**, first of all:

- Adaptive Firewall
- Adaptive Proxy
- Adaptive Gateway
- Application Firewall
- Application-level 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 Firewall**
- Web Application Security Device
- Web Application Proxy
- Web Application Shield
- Web Shield
- Web Security Firewall
- Web Security Gateway
- Web Security Proxy
- Web Intrusion Detection System
- Web Intrusion Prevention System

BREACH

WAF Identity Problem: Purpose

There are four aspects to consider:

1. **Audit device**
2. **Access control device**
3. **Layer 7 router/switch**
4. **Web Application Hardening tool**

The name (WAF) is overloaded. What about:

- **Web Intrusion Detection System?**
- **HTTP Security Monitoring?**

WAFEC

Short for Web Application Firewall Evaluation Criteria.

Project of the Web Application Security Consortium (webappsec.org).

It's an open project.

Virtually all WAF vendors on board (not enough users though).

WAFEC v1.0 released last year.

- New versions coming soon.

BREACH™

Part 2

ModSecurity

What is ModSecurity?

It is an **open source web application firewall**.

- **Most widely deployed web application firewall** according to Forrester Research.

That's not surprising because it is:

- Readily available.
- Full-featured.
- Stable and reliable.
- Well documented.
- Does what it says on the box.

History of ModSecurity

- Project started in 2002:
  - **“Wouldn’t it be nice if I had something to monitor what’s going on in my applications?”**
- Commercial support through Thinking Stone since 2004.
- Acquired by Breach Security in 2006.
  - **Breach Security pledges to support the open source nature of the project, adds resources.**
  - **Still going strong.**

BREACH

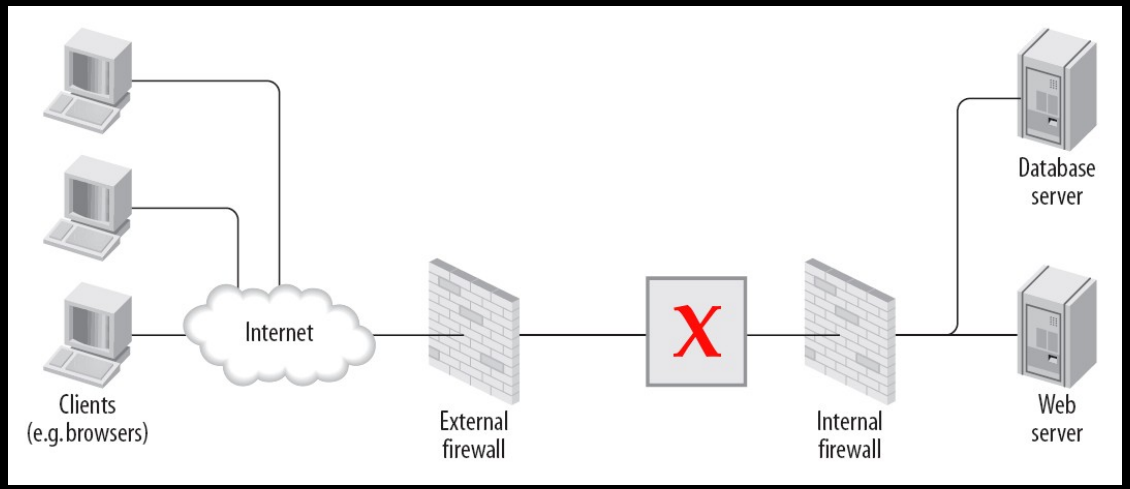
The Open Source Advantage

Four main points:

1. **Availability**
2. **Collaborative development**
3. **Transparency**
4. **Education**

Deployment Architectures

- **Embed** into your existing web servers.
- Deploy as a **network gateway** combining Apache working as reverse proxy with ModSecurity.



BREACH

Use Cases

1. HTTP intrusion detection and prevention.
2. Traffic logging.
3. Just-in-time patching.
4. Web application hardening.
   - For example, defending against the PDF UXSS vulnerability.

Security Models

1. Negative security
   - Easy to get started.
   - Trying to detect attacks.
   - Can be written by hand.
2. Positive security
   1. Must be tailored per application.
   2. But it only needs to determine what constitutes valid data.
   3. Virtual patches can be written by hand as they are simple. Automated learning required in all other cases.

ModSecurity Philosophy

- Make the WAF technology available to everyone.
- Nothing is done implicitly. You generally need to know what you're doing or use the pre-packaged rule sets.
  - Help users help themselves.
  - Do not surprise the user.
    - » Document everything.
    - » Tell it like it is.

ModSecurity Rule Language

- It's a simple event-based programming language.
  - Five processing phases, one for each major processing step.
  - Look at any part of the transaction.
  - Transform data to counter evasion.
  - Combine rules to form complex logic.
- Common tasks are easy, complex tasks are possible.

Advanced Features

- Persist information across requests.
  - You can create small databases of sorts.
- Support for anomaly-based rules.
- Support for sessions and application users.
- Log entire transactions or sessions.
  - Sanitise data before logging.
- Intercept file uploads.
- XML support (parse, validate, extract).

Rule Examples

Very simple (apply regex to input):

```
SecRule ARGS attack
SecRule ARGS|!ARGS:p attack
```

Different operator:

```
SecRule ARGS "@verifyByteRange 10,13,32-126"
```

Interesting:

```
SecRule REMOTE_ADDR "@rbl sc.surbl.org"
```

Real-life Example

Virtual patching example using the positive security approach:

```
<Location /apps/script.php>
    SecRule &ARGS "!@eq 1"
    SecRule ARGS_NAMES "!^statid$"
    SecRule ARGS:statID "!^\d{1,3}$"
</Location>
```

Rules should include meta-data, such as ID, revision, human-readable message, and so on.

Status

Stable version: **2.1.4**

Next major version: **2.5.0**

- **Parallel matching.**
- **GeoIP-based rules.**
- **Content injection.**
- **Credit-card number detection.**
- **Automated rule updates.**
- **PDF Universal XSS protection.**
- **Support for efficient and secure log centralisation.**
- **Full scripting support using Lua.**

Support for other web servers in **3.0.0**.

BREACH

BREACH

Part 3

Projects related to ModSecurity

ModSecurity Core Rules

Coherent set of rules designed to detect generic web application security attacks.

- Bundled with ModSecurity, but with a separate release cycle.
- Lead by Ofer Shezaf.

Design goals:

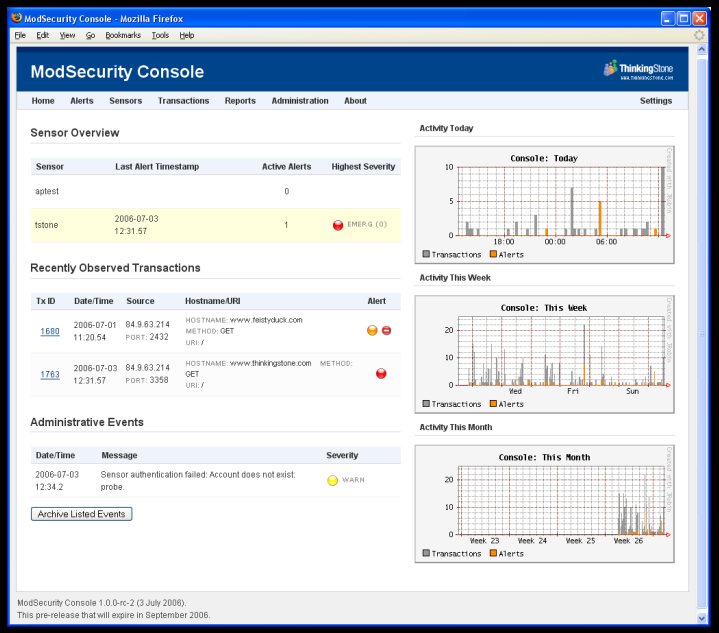
- Performance.
- Quality.
- Stability.
- Plug and Play.

Automated updates starting with ModSecurity 2.5.

ModSecurity Community Console

Self-contained application designed for alert aggregation, monitoring and reporting.

- Portable (Java).
- Free for up to 3 sensors.
- Not open source.

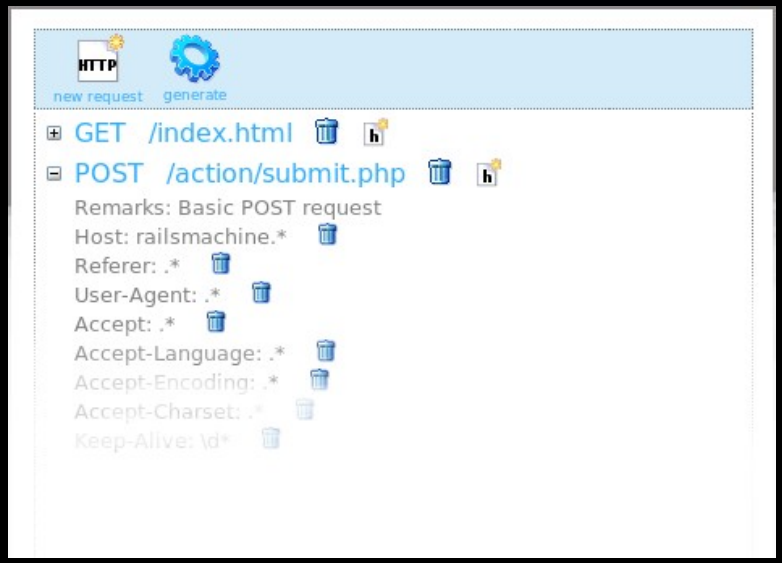


BREACH

REMO

A project to build a graphical rule editor for ModSecurity with a positive / whitelist approach.

- REMO stands for Rule Editor for ModSecurity.
- Community project run by Christian Folini.



BREACH

Distributed Open Proxy Honeypots

A network of open proxy sensors, each deployed with ModSecurity configured to log to the central server.

Goals:

- Observe what the bad guys are doing.
- Fine tune detection rules.
- WASC project (**webappsec.org**), run by Ryan Barnett.

Questions?

Thank you!

Ivan Ristic

ivan.ristic@breach.com

BREACH