BREACH

ModSecurity

The Open Source Web Application Firewall

Ivan Ristic

Chief Evangelist

Breach Security

Introduction
Ivan Ristic

- Background as **software developer** and **technical architect**.
- **Web application security** and **web application firewall** specialist.
- Author of **Apache Security** (O'Reilly, 2005).
- Author of **ModSecurity**.

modsecurity

BREACH

Case for Web Application Firewalls

- Web applications are written using loosely connected technologies and **inherently insecure**.
  - New issues are still being discovered.
- We need something reliable, for **monitoring** and **protection**, now.
- The term web application firewall has been **overloaded**… many times over.

Enter 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.

BREACH

History of ModSecurity

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

BREACH

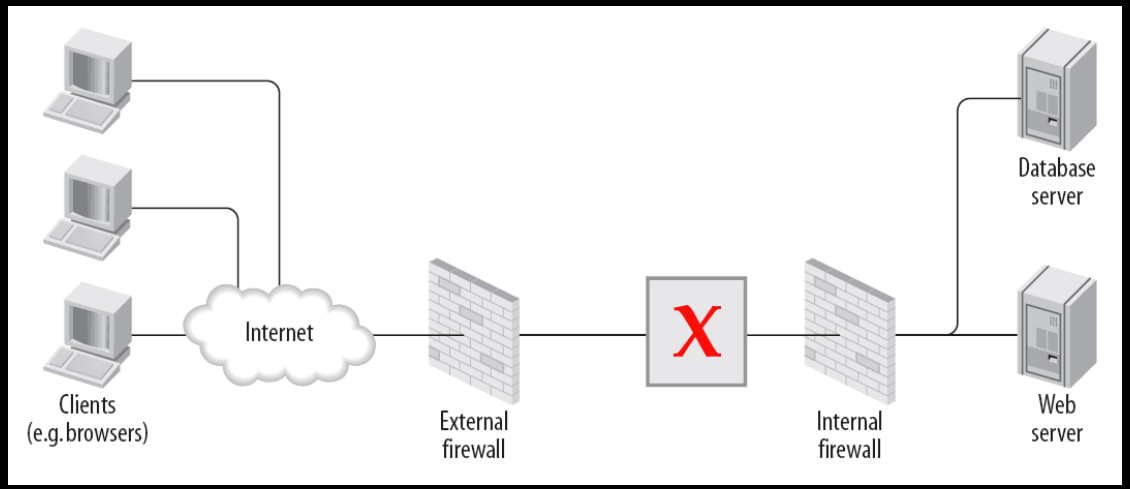
The Open Source Advantage

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

- Intrusion detection and prevention tool that speaks HTTP natively.
  - Negative security model.
  - Positive security model.
- Traffic logging.
- Just-in-time patching (a.k.a. virtual patching).
- Web application hardening.
  - For example, PDF XSS defence.

BREACH

ModSecurity Philosophy

- It's essentially a simple event-based programming language bundled with a bunch of parsers and transformation functions.
- Common tasks are easy, complex tasks are possible.
- Nothing is done implicitly. You generally need to know what you're doing or use pre-packaged rule sets.
- Document everything.

Interesting Features

- Five processing phases for every transaction.
- Flexible data transformation (mostly for anti-evasion).
- Stateful operation; supports any number of data “collections” (e.g. sessions, users, IP addresses).
- Support for anomaly scoring and event correlation.
- Understands sessions and users.
- Block by redirecting to Honeypot.
- XML support (parse, validate, and extract with XPath).
- Ability to easily extend the rule language.

ModSecurity 2.2+ Improvements

- Parallel (set-based) matching.
- GeoIP resolution.
- Performance improvements and optimisations (only relevant for very large rule sets but still).
- Modularity.
- Writing rules in C (and possibly using a scripting language – e.g. Lua).
- Support for any character encoding on input.
- Other interesting features: link rewriting, cookie protection, PDF XSS protection, etc.

Related Projects

- ModSecurity Core Rules
  - Coherent set of rules designed to address common web application security issues.
- ModSecurity Community Console
  - Alert aggregation and GUI.
  - Free for up to 3 sensors.
- Web Application Firewall Evaluation Criteria (WAFEC)
- Distributed Open Proxy Honeypots

BREACH

Questions?

Thank you!

Ivan Ristic

ivan.ristic@breach.com

BREACH