ĐỀ TÀI

TÌM HIỂU MOD_SECURITY VÀ DEMO TÍNH NĂNG

MSSV	Họ Tên	Lớp
10348551	Nguyễn Đức Lợi	DHTH4ATLT
10321271	Dương Hiễn Lãm	DHTH4BTLT
10356381	Nguyễn Phạm Hoàng Duy	DHTH4BTLT
10244481	Trịnh Minh Tiến	DHTH6C
09076271	Trần Công Chính	DHTH5C
11234421	Trần Anh Thiện	DHTH7B
11262121	Từ Kỷ	DHTH7B

Mục Lục

Chương 1: Giới thiệu 3

Chương 2: Modsecurity 4

I. Các khả năng của mod_security 5

II. Cài đặt 7

1. **Download mod_security:** 7
2. **Trước khi cài đặt** 7
3. **Giải nén** 7
4. **Biên dịch** 8
5. **Tích hợp modsecurity vào apache** 8
6. **Khởi động lại apache** 8

III. Cấu hình cơ bản 8

1. **File cấu hình** 8
2. **Turning Rule on and off** 8
3. **SecDefaultAction** 9

IV. Rules 9

1. **Xây dựng rules như thế nào?** 9
2. **Cấu trúc của rules** 11

2.1. Variables 11

2.2. Collections 12

2.3. Operators 12

2.4. Actions 13

V. Logging 14

1. **Debug Log** 14
2. **Audit logging** 15
3. **Tuỳ biến thông tin log** 16

VI. Xây dựng chính sách trên Modsecurity chống lại một số tấn công 16

1. **SQL Injection** 16

2. **XSS Attack** 17

3. **Brute force attacks** 20

4. **HTTP FingerPrinting** 21

Chương 3: Kịch bản Demo 22

Tài liệu tham khảo 24

Chương 1: Giới thiệu

Tường lửa ứng dụng web không phổ biến như tường lửa mạng, nhưng nó đã được đề cập đến trong các tin tức, bài báo và hội thảo bảo mật hiện nay. Các doanh nghiệp đã chấp nhận sử dụng công nghệ này bởi vì nó nâng cao an toàn web một cách đáng kể. Nhưng cấu hình, triển khai và duy trì công nghệ mới này không hề đơn giản. Để triển khai chúng thành công, bạn cần phải hiểu rõ các hoạt động của ứng dụng một cách đầy đủ và cấu hình các quy tắc tường lửa một cách cẩn thận. Hơn nữa, chi phí để mua được sản phẩm thương mại là rất đắt đỏ, chúng tôi khuyến khích bắt đầu sử dụng với các sản phẩm mã nguồn mở, chẳng hạn ModSecurity, do đó bạn có thể đưa ra quyết định, nếu giải pháp này là thích hợp với ngân sách và môi trường mạng của bạn.

Trong vài năm gần đây, các lỗ hổng ứng dụng web đã trở thành mối đe dọa lớn nhất trong môi trường công nghệ thông tin. Theo cơ sở dữ liệu lỗ hổng mã nguồn mở (OSVDB), các mối đe dọa ứng dụng web chiếm hơn 50% tất cả các lỗ hổng trong năm 2010. Bạn không phải một chuyên gia trong lĩnh vực IT để cấu hình các ứng dụng web đã trở nên sử dụng rộng rãi trong cuộc sống cũng như trong kinh doanh. Do đó, bảo mật các ứng dụng web đã trở thành một vấn đề quan trọng nhất mà bạn cần phải chú ý như người dùng cuối hoặc một người kinh doanh.

Tường lửa ứng dụng web (WAF) là một dạng tường lửa nhằm lọc các lưu lượng HTTP dựa trên một tập quy tắc. Nó kiểm tra ở mức ứng dụng vì vậy nó thường đi kèm như một thiết bị hoặc một mô đun trong máy chủ, có chức năng xác định và ngăn chặn các tấn công web phổ biến như cross-site scritpting (XSS) và SQL injection bằng việc tùy chỉnh các quy tắc. Do đó, việc tùy chỉnh các quy tắc này là rất đáng quan tâm và yêu cầu bảo trì cao.

Có nhiều cách để bảo vệ ứng dụng web, chẳng hạn như thực hiện một đoạn mã an toàn, quản lý cấu hình an toàn, thực hiện đánh giá lỗ hổng và triển khai tường lửa ứng dụng web, nhưng không có một giải pháp nào là hoàn hảo, việc sử dụng tường lửa ứng dụng web chỉ là một phương thức bảo mật ứng dụng. Kỹ thuật này tương đối mới so với các công nghệ khác, nhưng nó có thể trở thành một giải pháp hữu hiệu khi bạn cấu hình và sử dụng nó đúng cách.

Trong bài giới thiệu này, Modsecurity sẽ được sử dụng để minh hoạ làm thế nào bảo mật ứng dụng web sử dụng WAF. Modsecurity bảo vệ các ứng dụng web từ nhiều cuộc tấn công và cho phép giám sát lưu lượng HTTP với sự can thiệp không nhiều của cơ sở hạ tầng hiện có. Nó là một mô đun mã nguồn mở của ứng dụng máy chủ web Apache và nó đã được bảo trì bởi SpiderLabs, Trustwave. Từ khi nó là một sản phẩm mã nguồn mở, nó mặc nhiên là miễn phí và nhiều người dùng sử dụng đã góp phần cải thiện và duy trì sản phẩm này.

WAF không phải là một công cụ mà chỉ nhằm ngăn chặn các hoạt động nguy hiểm tại lớp ứng dụng. Nó cũng có thể được sử dụng để phân tích và phát hiện các lưu lượng nguy hiểm tấn công vào các ứng dụng. Do đó, phần sau của loạt bài viết này sẽ minh họa làm thế nào phân tích các tấn công web phổ biến sử dụng WAF để phát hiện và ghi lại các khả năng cùng với nhật ký máy chủ Web Apache.

Chương 2: Modsecurity

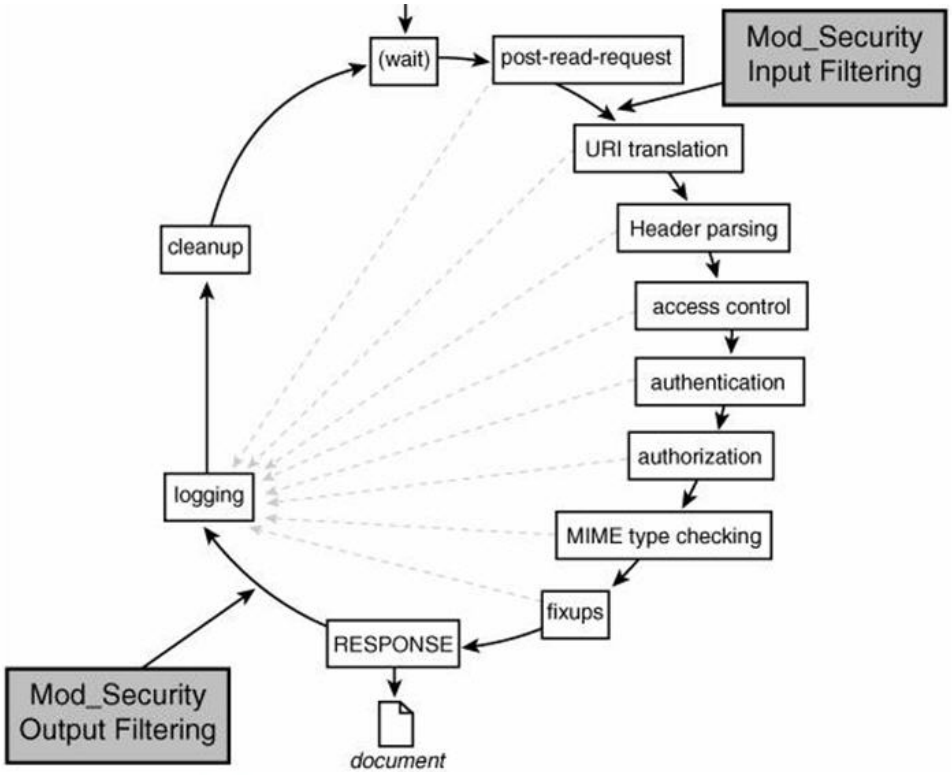
Mod_security là một opensource web application firewall được Ivan Ristic phát triển dành cho Apache Web Server. Ivan Ristic là tác giả quyển sách.Ông là một người có rất nhiều kinh nghiệm trong bảo vệ Apache Web Server. Ông đã có nhiều thời gian nghiên cứu Web Application Security, Web Intrusion Detection, và Security Patterns. Trước khi chuyển sang lĩnh vực security, Ivan đã có nhiều năm làm việc như một developer, system architect, technical director trong phát triển phần mềm. Ông là người sáng lập ra công ty ThinkingStone làm các dịch vụ liên quan đến web application security.

Hiện tại mod_security sử dụng giấy phép GPL, hoàn toàn miễn phí. Ngoài ra nếu muốn có sự hỗ trợ thì bạn có thể mua nó tại công ty ThinkingStone của ông (http://www.thinkingstone.com)

I. Các khả năng của mod_security

- Request filtering : tất cả các request gửi đến web server đều được phân tích và cản lọc (filter) trước khi chúng được đưa đến các modules khác để xử lý.
- Anti-evasion techniques: paths và parameters được chuẩn hoá trước khi phân tích để chống evasion techniques. Kỹ thuật này sẽ được thảo luận ở phần sau.
- Understanding of the HTTP protocol: mod_security là web application firewall nên nó có khả năng hiểu được HTTP protocol. Mod_security có khả năng cản lọc dựa trên các thông tin ở HTTP Header hay có thể xem xét đến từng parameters hay cookies của các requests ...vv
- POST payload analysis: ngoài việc cản lọc dựa trên HTTP Header, mod_security có thể dựa trên nội dung (payload) của POST requests.
- Audit logging : mọi requests đều có thể được ghi lại (bao gồm cả POST) để chúng ta có thể xem xét sau nếu cần.
- HTTPS filtering: mod_security có thể phân tích HTTPS.
- Compressed content filtering: mod_security sẽ phân tích sau khi đã decompress các request data.

Quá trình xử lý các request của Apache và mod_security



Modsecurity cho phép bạn đặt rule tại một trong năm thời điểm trong chu kỳ xử lý của Apache như sau:

Phase Request Header: *rule* được đặt tại đây sẽ được thực hiện ngay sau khi Apache đọc request header, lúc này phần request body vẫn chưa được đọc.

Phase Request Body: đây là thời điểm các thông tin chức năng chung đưa vào vào được phân tích và xem xét, các rule mang tính application-oriented thương được đặt ở đây. Ở thời điểm này bạn đã nhận đủ các request argument và phần request body đã được đọc.**Modsecurity** hỗ trợ ba loại mã hoá request body

+ application/x-www-form-urlencoded dùng để truyền form dữ liệu

+ multipart/form-data dùng để truyền file

+ text/xml dùng để phân tich dữ liệu XML

Phase Response Header: đây là thời điểm ngay sau khi phần response header được gửi trả về cho client. Bạn đặt rule ở đây nếu muốn giám sát quá trình sau khi phần response được gửi đi.

Phase Response Body: đây là thời điểm bạn muốn kiểm tra những dữ liệu HTML gửi trả về

Phase logging: đây là thời điểm các hoạt động log được thực hiện, các rules đặt ở đây sẽ định rõ việc log sẽ như thế nào, nó sẽ kiểm tra các error message log của Apache. Đây cung là thời điểm cuối cùng để bạn chặn các connection không mong muốn, kiểm tra các response header mà bạn không thể kiểm tra ở phase 3 và phase 4.

II. Cài đặt

1. **Download mod_security:**

   #wget
   http://www.modsecurity.org/download/modsecurity-apache_2.5.13.tar.gz

2. **Trước khi cài đặt**

   Cần các thư viện apxs, libxml2 và cần file mod_unique_id.so

   Cài đặt thư viện

   #yum install httpd-devel (cài apxs)

   #yum install libxml2-devel

   Thêm dòng sau vào file http.conf (file nằm trong /etc/httpd/conf) dòng sau

   LoadModule unique_id_module modules/mod_unique_id.so

   (Chú ý: thêm vào đoạn có nhiều LoadModule đầu dòng)

3. **Giải nén**

   #tar xzvf modsecurity-apache_2.5.13.tar.gz

4. **Biên dịch**

Tại thư mục apache2 gõ các lệnh sau

#./configure

#make

#make install

5. **Tích hợp modsecurity vào apache**

Thêm dòng sau vào file httpd.conf

LoadModule security2_module modules/mod_security2.so

6. **Khởi động lại apache**

#service httpd restart

III. Cấu hình cơ bản

1. **File cấu hình**

Modsecurity là application firewall thuộc loại rules-based, nghĩa chúng ta cần thiết lập các luật (rules) để mod_security hoạt động. Các rules này được thể hiện dưới dạng các chỉ thị (directives) và có thể đặt trực tiếp trong file cấu hình Apache (thông thường là httpd.conf).

Ngoài ra có thể đặt các cấu hình này vào một file riêng, chẳng hạn modsecurity.conf trong thư mục conf.d và sau đó chúng ta cần thêm vào httpd.conf

Include conf.d/modsecurity.conf

(mặc định trong httpd.conf đã có dòng include conf.d/*.conf với dòng này nó sẽ thực hiện tất cả các file có phần mở rộng là .conf)

2. **Turning Rule on and off**

Theo mặc định thì rule engine bị disable. Để kích hoạt modsecurity ta cần thêm chỉ thị sau vào file cấu hình

SecRuleEngine On

Directive này dùng để điều khiển rule engine, chúng ta có thể sử dụng các tuỳ chọn là On, Off hoặc DynamicOnly.

Off : Vô hiệu hoá modsecurity

DetectionOnly : Khi nó phù hợp một luật nào đó thì nó cũng không thực hiện bất kì action nào. (nó rất có ích trong trường hợp muốn test một luật nào đó mà không muốn nó block bất kì request nào có vấn đề với luật)

On : Các rules của modsecurity được áp dụng cho tất cả các nội dung

3. **SecDefaultAction**

Dùng để tạo các action mặc định. Khi tạo 1 luật mà không chỉ rõ hành động cho luật đó nó sẽ thực hiện hành động mặc định

Ví dụ: *SecDefaultAction "phase:2,deny,log,status:403"*

IV. Rules

1. **Xây dựng rules như thế nào?**

HTTP request

GET /documentation/index.html HTTP/1.1

Host: www.modsecurity.org

User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.0.1) ecko/20060124 Firefox/1.5.0.1

Accept: text/xml,application/xml,application/xhtml+xml, text/html;q=0.9,text/plain;q=0.8,image/png,/*;q=0.5*

Accept-Language: en-us,en;q=0.5

Accept-Encoding: gzip,deflate

Accept-Charset: ISO-8859-1,utf-8;q=0.7,;q=0.7*

Keep-Alive: 300

Connection: keep-alive

Referer: http://www.modsecurity.org/index.php

Cookie:__utmz=129890064.1139909500.1.1.utmccn=(direct)|utmcsr=(direct)|utmcmd=(none);