

Guida All'installazione di ModSecurity on Nginx

Lorenzo Mauriello

December 5, 2022

Abstract

Guida completa all'installazione di ModSecurity su Nginx, il flusso di operazioni è valido sui principali sistemi linux come Ubuntu e CentOS. Programmi necessari:

1. Ubuntu 20.04 lts (focal).
2. Nginx 1.18.0.
3. ModSecurity3.
4. ModSecurity-nginx.
5. ModSecurity-crs 1.0.2.

Introduction

Prima di procedere bisogna con l'installazione e la configurazione delle singole componenti bisogna scaricare le dipendenze necessarie:

```
1 $ sudo apt-get install bison build-essential ca-certificates \  
2 curl dh-autoreconf doxygen flex gawk git iputils-ping \  
3 libcurl4-gnutls-dev libexpat1-dev libgeoip-dev libltdb-dev \  
4 libpcre3-dev libpcre++-dev libssl-dev libtool libxml2 libxml2-dev \  
5 libyajl-dev locales lua5.3-dev pkg-config wget zlib1g-dev zlibc
```

Il passaggio successivo all'installazione delle dipendenze per la compilazione, possiamo installare nginx così da preparare l'ambiente di lavoro:

```
1 $ sudo apt-get update  
2 $ sudo apt-get upgrade  
3 $ sudo apt-get install nginx
```

Questo completa il setup preliminare, è importante sottolineare l'installazione delle dipendenze altrimenti si avranno problemi in fase di compilazione dei moduli software. Ancora, bisogna creare una cartella dove andranno tutti i download, in questo caso si utilizzerà la cartella */Home*.

ModSecurity download and compile

Portiamoci nella */Home* e scarichiamo il modulo ModSecurity:

```
1 $ git clone https://github.com/SpiderLabs/ModSecurity  
2 $ cd ModSecurity  
3 $ git checkout -b v3/master origin/v3/master  
4 $ git submodule init  
5 $ git submodule update
```

Una volta che il download è completo passiamo a compilare il modulo:

```
1 $ ./build.sh  
2 $ ./configure
```

I comandi appena eseguiti verificano la configurazione di sistema, se tutto è andato a buon fine possiamo procedere alla compilazione del modulo:

```
1 $ make  
2 $ sudo make install
```

Attenzione ai comandi **make** (in modo opzionale `-j <number cpu>` si possono incrementare le performance), questi richiedono molto tempo per essere eseguiti. Durante i periodi di attesa si possono scaricare i moduli restanti.

Errori comuni

Errore 1 Potrebbe capitare che i comandi iniziali di configurazione non funzionino, questo perchè le dipendenze sono state installate male o necessitano di aggiornamento.

Errore 2 Il comando **make install** fallisce [permission denied], ricordare di inserire sudo se non si hanno i permessi di root.

ModSecurity-Nginx download and compile

In questa sezione andiamo a scaricare e compilare l'interfaccia di comunicazione che permette a ModSecurity di interagire con Nginx:

```
1 $ cd ..
2 $ git clone https://github.com/SpiderLabs/ModSecurity-nginx
```

A questo punto dobbiamo controllare la versione di Nginx che abbiamo sul nostro sistema operativo (nel nostro caso sarà la versione 1.18.0):

```
1 $ nginx -v
```

procedere scaricando tramite wget la versione indicata dal sistema operativo per poi poter compilare gli shared object che ci servono:

```
1 $ wget http://nginx.org/download/nginx-1.18.0.tar.gz
2 $ tar -zxvf nginx-1.18.0.tar.gz
3 $ cd nginx-1.18
```

ora il setup è pronto, siamo nella nostra cartella di Nginx, ma ancora non possiamo procedere a compilare. La cosa che adesso dobbiamo verificare è capire quali moduli Nginx sono installati sul nostro sistema operativo:

```
1 $ nginx -V
```

a questo punto dobbiamo copiare tutti i moduli che ci escono a schermo come nell'esempio *ricordarsi di inserire figura d'esempio* dopo di che si procede:

```
1 $ ./configure [moduli precedentemente identificati]
   --add-dynamic-modules=../ModSecurity-nginx
```

Durante questa fase di configurazione si verificheranno quasi certamente degli errori dovuti a librerie mancanti (nel mio caso sono state installate le componenti libxslt-dev e libgd-dev). Dopo ogni errore bisogna ripetere il comando di configurazione fin quando non si termina con successo l'esecuzione. Solo ora è possibile creare i moduli:

```
1 $ make modules
```

Fatto questo otteniamo un file `ngx_http_modsecurity_module.so` che dovrà essere copiato all'interno dei moduli validi di Nginx:

```
1 $ sudo mkdir /etc/nginx/modules
2 $ sudo cp ngx_http_modsecurity_module.so /etc/nginx/modules
```

dobbiamo quindi inserire all'interno del file di configurazione di nginx il modulo appena creato ed istanziato, per far ciò:

```
1 $ sudo nano /etc/nginx/nginx.conf
```

io nello specifico utilizzerò sempre nano, ma un qualunque editor equivalente come vim va più che bene. Entrati nel file di configurazione bisogna inserire all'inizio del file (prima della classe event ma dopo gli include) il caricamento del modulo:

```
1 $ load_module /etc/nginx/module/ngx_http_modsecurity_module.so;
```

In questo momento modsecurity non è ancora in grado di interagire con nginx, dobbiamo ancora creare un riferimento alle **Core Rule Set** che sono messe a disposizione contro i principali attacchi informatici di livello applicativo.

Core Rule Set e attivazione di ModSecurity

A questo punto il prossimo passo è portarci nella cartella **/Home** e scaricare le modsecurity-crs:

```
1 $ git clone https://github.com/coreruleset/coreruleset modsecurity-crs
2 $ cd modsecurity-crs
```

bisogna adesso attivare le configurazioni di base che in questo momento sono riportate come semplici esempi:

```
1 $ mv crs-setup.conf.example crs-setup.conf
2 $ cd rules
3 $ mv REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf.example
   REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf
4 $ cd ..
5 $ cd ..
```

spostiamo le CRS appena scaricate in modo che queste possano essere lette pubblicamente:

```
1 $ mv modsecurity-crs /usr/local/
```

si inseriscono dei file di configurazione all'interno di nginx:

```
1 $ mkdir -p /etc/nginx/modsec
2 $ cp /ModSecurity/unicode.mapping /etc/nginx/modsec
3 $ cd ModSecurity
4 $ mv modsecurity.conf-recommended modsecurity.conf
5 $ cp modsecurity.conf /etc/nginx/modsec
6 $ sudo nano /etc/nginx/modsec/modsecurity.conf
7 # cambiare all'interno del file selezionato nel campo SecRuleEngine On
8 $ cd /etc/nginx/modsec/
9 $ sudo nano main.conf
10 # inserire all'interno di questo file gli include di tutti i file di configurazione
11 '-----
12 Include /etc/nginx/modsec/modsecurity.conf
13 Include /usr/local/modsecurity-crs/crs-setup.conf
14 Include /usr/local/modsecurity-crs/rules/*.conf
15 -----'
```

> ora per concludere la fase di configurazione di nginx dobbiamo settare la possibilità di attivare o disattivare modsecurity in modo molto semplice:

```
1 $ sudo nano /etc/nginx/sites-available/default
2 # scrivere all'interno di questo file nella classe server{}
3 '-----
4 modsecurity on;
5 modsecurity_rules_file /etc/nginx/modsec/main.conf;
6 -----'
```

a questo punto possiamo riavviare nginx "**systemctl restart nginx**" e verifichiamo che funzioni correttamente. Se tutto è stato fatto correttamente mod security è attivo e possiamo proteggere la nostra applicazione con successo.