Stefano Busnelli (IlGuru)

PROXY ANONYMIZER

6 April 2018

Preambolo

Tempo fa sul forum si parlava di far sparire la pubblicità da qualche sito web per poterlo consultare senza banner invadenti che saltano davanti agli occhi causando convulsioni e nausea. Dissi che avevo ovviato al problema configurando su uno dei miei raspberry un proxy anonymizer, ma all'epoca non avevo fatto abbastanza ordine per poterlo rendere disponibile.
Ciò che segue è la descrizione del progetto con tanto di link per utilizzare liberamente quanto ho elaborato. Come ho detto funziona su un raspberry, quindi con raspbian e architettura armv7l, ma ho creato immagini docker che ho testato anche su una macchina virtuale Oracle Virtalbox dove ho installato una debian con architettura x86_64.

Se l'host che ospiterà i docker avrà una di queste due architetture gli script sceglieranno le images da utilizzare, altrimenti aprite un ticket su gitlab e crosscompilerò. Probabilmente funziona anche sotto windows, ma li i veri problemi sono altri...in riferimento alle cose 'dockerose' utilizzerò il loro termine originale anglosassone invece che la traduzione italiana, quindi docker images, services, volumes, stack, deploy ecc...

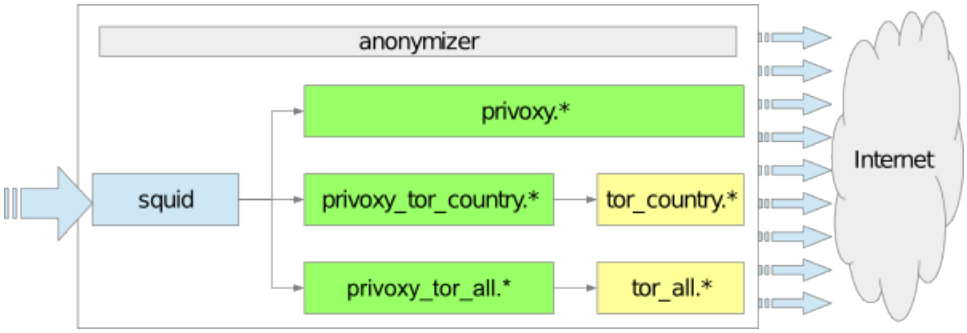
Perchè docker?
Perchè è un sistema di virtualizzazione fighissimo che occupa pochissimo spazio e pochissime risorse, contrariamente a quanto farebbe una qualunque virtual machine basata su virtualhost o vmware. Le immagini si creano come normali macchine linux con i comandi tipici di installazione quali apt ecc... vengono salvate in cloud e producono container che possono essere scalati e ed inseriti in cluster giganteschi, tutto gestito dall' infrastruttura docker. Insomma rispetto al solito uso di macchine virtuali, la fantascienza.

Descrizione:

Questo anonymizer è basato su docker ed ogni image utilizza docker-gen per generare automaticamente i file di configurazione dei vari servizi tramite alcuni templates scritti in go-lang. Le docker images sono disponibili su docker cloud, sono compilate per architetture armv7l (raspberry PI 3) e x86_64 ma possono essere ricompilate usando lo script:

```
# . bin/build-all.sh
```

Le docker images vengono schierate in una stack chiamata 'anonymizer' che rende disponibili 6 services collegati tra di loro in 3 catene di parent proxies organizzati su 3 livelli in cui sono presenti i servizi squid, privoxy and tor:



anonymizer.png

- squid

In questa image è installato squid.

Mette a disposizione una proxy in ascolto sulla porta 3128 che risponde alle richieste del browser, fa cache ed è configurato per rimuovere dagli header inviati dal browser tutte le informazioni utilizzate dai siti web per tracciare l'utente come tipologia di browser o sistema operativo, risoluzione del monitor, indirizzi ip interni ecc..

Ci sono due access control list basate sulla direttiva dstdom_regex che permettono a squid di decidere su quale tipo di catena di parent proxy inoltrare le richieste per raggiungere i siti web che devono essere inseriti in 2 file utilizzando la sintassi delle espressioni regolari.

- I siti web che devono essere raggiunti direttamente da privoxy senza passare per la rete tor, vanno inseriti nel file /etc/squid/pool_no_tor, che è esposto nella macchina host.

Le richieste inoltrate in queste catene di proxy appariranno come provenienti dal vero ip address del browser, quindi saranno inseriti solo siti sicuri.

Esempio di pool_no_tor:

```
(^|\.)google\.(\w)*($|.*)
(^|\.)youtube\.(\w)*($|.*)
(^|\.)googleapis\.(\w)*($|.*)
(^|\.)gstatic\.(\w)*($|.*)
```

- I siti che devono essere raggiunti tramite le catene *_tor_country devono essere inseriti nel file /etc/squid/pool_tor_country che viene esposto nella macchina host.

Le richieste inoltrate in queste catene di proxy appariranno come provenienti dal paese selezionato nel file di configurazione dello stack ma non dal vero ip address del browser.

Esempio di pool_tor_country:

```
(^|\.)facebook\.(\w)*($|.*)
(^|\.)fbcdn\.(\w)*($|.*)
(^|\.)fbsbx\.(\w)*($|.*)
(^|\.)microsoft\.(\w)*($|.*)
(^|\.)amazon\.(\w)*($|.*)
```

    - Tutte le richieste verso i siti web che non sono inseriti nei due file verranno inoltrate nelle catene *_tor_all ed appariranno come provenienti da qualsiasi parte del pianeta.

- privoxy

in questa image è installato privoxy. Privoxy è un proxy che non fa cache dotato di funzionalità avanzate di filtri che aumentano la privacy, modifica le pagine web e gli header HTTP, controlla gli accessi, e rimuove pubblicità e spazzatura dalle pagine web. E' configurato per accettare richieste da squid ed inoltrarle al sito di destinazione o attraverso la rete tor. Questa image genera 3 tipi di services:

    - privoxy

    parent proxy di squid che inoltra le richieste direttamente al sito di destinazione mostrando il vero indirizzo ip del router.

    - privoxy_tor_country

    parent proxy di squid che inoltra le richieste attraverso il service tor_country

    - privoxy_tor_all

    parent proxy di squid che inoltra le richieste attraverso il service tor_all

- tor

In questa image è installato il client tor.

Tor è una rete di proxy che aiutano a difendersi contro il tracciamento e l'analisi del traffico web, una forma di sorveglianza che minaccia la privacy e la libertà personale. Nasconde il reale ip address del client quindi i siti di destinazione vedranno il traffico come proveniente da un punto del mondo dove la rete tor (dopo vari salti) farà uscire la connessione.

Questa image genera 2 tipi di services:

    - tor_country

    Parent proxy di privoxy_tor_country che utilizza solo nodi tor del paese selezionato dalla variabile E_EXIT_NODES nel file /opt/anonymizer/etc/docker-compose-*.yml con cui viene generato lo stack anonymizer. Attualmente E_EXIT_NODES=it quindi solo nodi tor italiani.

    - tor_all

Parent proxy di privoxy_country che utilizza nodi tor di tutto il mondo.

Solamente squid è esposto all'esterno tramite la porta 3128, privoxy è solamente parent proxy di squid e tor è parent proxy di privoxy. I services Privoxy e Tor possono essere moltiplicati tramite docker compatibilmente con le capacità del sistema host, tutti i file di configurazione verranno automaticamente rigenerati ed i servizi verranno fatti ripartire. Nello stack docker ci sono 3 volumes statici in cui le principali cartelle usate da squid vengono montate nel filesystem dell'host per rendere disponibili i log ed i file di configurazione:

```
# docker volume ls | grep anonymizer
local               anonymizer_squid_etc
local               anonymizer_squid_log
local               anonymizer_squid_spool
```

squid_etc:

La cartella /etc/squid è montata in /var/lib/docker/volumes/anonymizer_squid_etc/_data

squid_log:

La cartella /var/log/squid è montata in /var/lib/docker/volumes/anonymizer_squid_log/_data

squid_spool:

La cartella /var/spool/squid è montata in /var/lib/docker/volumes/anonymizer_squid_spool/_data

Maggiori iniformazioni sui 3 volumes:

```
# docker volume inspect anonymizer_squid_etc
```

Grazie a questi 3 volumes statici quando squid viene fatto ripartire i file rimangono salvati sull' host cosicchè tutta la cache ed i log vengono mantenuti.

Installazione:

1. Installazione di docker

```
# apt install docker-ce
```

2. prelievo dell' anonymizer:

```
# git clone git@gitlab.com:StefanoBusnelli/anonymizer.git
```

```
# cd anonymizer
# . install.sh
```

Istruzioni:

Start proxy:

```
# .  /opt/anonymizer/bin/start.sh
```

Stop proxy:

```
# .  /opt/anonymizer/bin/stop.sh
```

Restart proxy:

```
# .  /opt/anonymizer/bin/restart.sh
```

Update services:

```
# .  /opt/anonymizer/bin/update_services.sh
```

List services:

```
# docker stack services anonymizer
ID              NAME                                MODE            REPLICAS    IMAGE
gykb3qutthsg    anonymizer_tor_all                  replicated      5/5         busnellist
hy80orv1p4d9    anonymizer_privoxy                  replicated      2/2         busnellist
iiw2fc2u2qle    anonymizer_squid                    replicated      1/1         busnellist
vuc7cbbhl3bm    anonymizer_tor_country              replicated      5/5         busnellist
wrb5bt71mh38    anonymizer_privoxy_tor_all          replicated      5/5         busnellist
zm06ue10t952    anonymizer_privoxy_tor_country      replicated      5/5         busnellist
```

Aumento/Diminuzione dei services:

E' possibile scalare i servizi secondo le necessità ed il file squid.conf verrà automaticamente aggiornato. L'unica restrizione (che mi pare ovvia) è che i proxy tor siano in numero uguale ai privoxy dello stesso tipo:

```
# docker service scale anonymizer_privoxy=3
anonymizer_privoxy scaled to 3
overall progress: 2 out of 3 tasks
1/3: starting  [============================================>      ]
2/3: running   [==================================================>]
3/3: running   [==================================================>]
```

```
# docker service scale anonymizer_privoxy_tor_all=10
anonymizer_privoxy_tor_all scaled to 10
overall progress: 5 out of 10 tasks
1/10: running   [==================================================>]
2/10: running   [==================================================>]
3/10: running   [==================================================>]
4/10: running   [==================================================>]
5/10: running   [==================================================>]
6/10: starting  [============================================>      ]
7/10: starting  [============================================>      ]
8/10: starting  [============================================>      ]
9/10: starting  [============================================>      ]
10/10: starting  [============================================>      ]

# docker service scale anonymizer_tor_all=10
anonymizer_tor_all scaled to 10
overall progress: 5 out of 10 tasks
1/10: running   [==================================================>]
2/10: running   [==================================================>]
3/10: running   [==================================================>]
4/10: running   [==================================================>]
5/10: running   [==================================================>]
6/10: starting  [============================================>      ]
7/10: ready     [=====================================>             ]
8/10: preparing [==================================>                ]
9/10: preparing [==================================>                ]
10/10: preparing [==================================>                ]
```

Gestione di squid:

Lista delle opzioni:

```
# docker exec $(docker ps | grep anonymizer_squid | cut -c -12) squidclient mgr:menu
```

Utilizzo della memoria:

```
# docker exec $(docker ps | grep anonymizer_squid | cut -c -12) squidclient mgr:mem
```

Statistiche sui parent proxy utilizzati

```
# docker exec $(docker ps | grep anonymizer_squid | cut -c -12) squidclient mgr:server_
```

Database e statistiche dei siti richiesti:

```
# docker exec $(docker ps | grep anonymizer_squid | cut -c -12) squidclient mgr:netdb
```

Statistiche sull'utilizzo della cache:

```
# docker exec $(docker ps | grep anonymizer_squid | cut -c -12) squidclient mgr:storedi
```

Riferimenti

- docker
- squid
- privoxy
- tor project
- docker-gen
- go-lang

Estratto da "http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Ilguru:proxy-anonymizer-2"