How To LEMP su CentOS 7

How To: Installare Linux, Nginx, MySQL, PHP 7 (LEMP) su CentOS 7

Un ambiente LEMP a differenza dell'ambiente LAMP (Linux, Apache, MySQL, PHP) differisce per il web server ENGINX.

In questo HOW TO andremo a vedere come installare PHP 7 FPM con Nginx per avere le massime prestazioni.

Il primo passo da fare è disabilitare il SELINUX, la cosa migliore sarebbe configurarlo ad hoc, ma questa operazione va eseguite prima della messa in esercizio del server.

Primo step installare Nginx

Come primo passaggio installiamo il repository EPEL e IUS. Utilizziamo il comodissimo script messo a disposizione sul sito ius.io per fare prima. Il curl ci aiuta a scaricare il setup.

```
curl 'https://setup.ius.io/' -o setup-ius.sh
```

ora eseguiamo lo script:

```
bash setup-ius.sh
```

Ora possiamo installare NGINX

```
yum install nginx
```

avviamo il webserver con il comando systemctl

```
systemctl start nginx
```

ora possiamo provare puntando nel nostro browser

```
http://indirizzo_ip_del_server/
```

Se vedi questa pagina allora possiamo abilitare NGINX all'avvio del sistema,

al BOOT:

```
systemctl enable nginx
```

Secondo Step installare MySQL

Abbiamo due possibilità, installare MySQL o MariaDB, sono la stessa cosa nati dallo stesso adre **Ulf Michael Widenius** noto anche come **Monty.**

In questa guida opteremo per MySQL, il comando è il seguente:

```
yum install mysql-server mysql
```

Ora passiamo alla prima configurazione del nostro RDBMS: start del demone:

```
service mysqld status
```

e poi messa in sicurezza di base:

```
mysql_secure_installation
```

Siamo pronti per abilitare anche MySQL al boot:

```
systemctl enable mysqld
```

Terzo Step installazione di PHP-FPM 7

Ora passiamo all'installazione dei PHP-FPM (FastCGI Process Manager) 7, l'ultima versione del php disponibile ad oggi eseguito sulla porta 9000:

```
yum install php70u-fpm-nginx php70u-cli php70u-mysqlnd
```

installato apriamo il file di configurazione e sostituiamo l'utente e i l gruppo d'esecuzione:

```
vim /etc/php-fpm.d/www.conf
```

```
; When POSIX Access Control Lists are supported you can set them using
; these options, value is a comma separated list of user/group names.
; When set, listen.owner and listen.group are ignored
;listen.acl_users = apache,nginx
;listen.acl_users = apache
listen.acl_users = nginx
;listen.acl_groups =
```

a questo punto riavviamo creiamo un vhosts, per prima cosa per tenere in ordine il nostro ambiente posizioniamo i file dei virtual hosts in una directory:

```
mkdir /etc/nginx/sites-available
```

passiamo al file nginx.conf l’istruzione di leggere il contenuto della nuova direcotry

```
vim /etc/nginx/nginx.conf
```

aggiungendo la riga

```
include /etc/nginx/sites-enabled/*;
```

```
;server {
listen 81.127.13.234:80;
server_name stat.lbit-solution.it;
location / {
try_files $uri $uri/ =404;
root /var/www/vhosts/lbit-solution.it/stat.lbit-solution.it/;
index index.php index.html index.htm;
}

error_page 404 /404.html;
error_page 500 502 503 504 /50x.html;
location = /50x.html {
root /var/www/vhosts/lbit-solution.it/stat.lbit-solution.it/;
}

location ~ \.php$ {
root /var/www/vhosts/lbit-solution.it/stat.lbit-solution.it/;
fastcgi_pass 127.0.0.1:9000;
fastcgi_index index.php;
fastcgi_param SCRIPT_FILENAME /var/www/vhosts/lbit-solution.it/stat.lbit-
solution.it$fastcgi_script_name;
include fastcgi_params;
}
}

server {
listen 81.127.13.234:443 ssl;
server_name stat.lbit-solution.it;

### SSL cert files ###
```

```
ssl_certificate /var/www/vhosts/lbit-solution.it/ssl/stat.lbit-
solution.it.crt;
ssl_certificate_key /var/www/vhosts/lbit-solution.it/ssl/stat.lbit-
solution.it.key;

### Add SSL specific settings here ###

ssl_protocols SSLv3 TLSv1 TLSv1.1 TLSv1.2;
ssl_ciphers RC4:HIGH:!aNULL:!MD5;
ssl_prefer_server_ciphers on;
keepalive_timeout 60;
ssl_session_cache shared:SSL:10m;
ssl_session_timeout 10m;

### SSL log files ###
access_log /var/www/vhosts/lbit-solution.it/logs/stat.lbit-solution.it.ssl-
access.log;
error_log /var/www/vhosts/lbit-solution.it/logs/stat.lbit-solution.it.ssl-
error.log;

location / {
try_files $uri $uri/ =404;
root /var/www/vhosts/lbit-solution.it/stat.lbit-solution.it/;
index index.php index.html index.htm;
}

error_page 404 /404.html;
error_page 500 502 503 504 /50x.html;
location = /50x.html {
root /var/www/vhosts/lbit-solution.it/stat.lbit-solution.it/;
}

location ~ \.php$ {
root /var/www/vhosts/lbit-solution.it/stat.lbit-solution.it/;
fastcgi_pass 127.0.0.1:9000;
fastcgi_index index.php;
fastcgi_param SCRIPT_FILENAME /var/www/vhosts/lbit-solution.it/stat.lbit-
solution.it$fastcgi_script_name;
include fastcgi_params;
}
}
```

Prepariamo le direcotry del virtual hosts:

```
mkdir -p /var/www/vhosts/lbit-solution.it/{ssl,logs,httpdocs,stat.lbit-
solution.it}
```

E ora riavviamo php-fpm e Nginx

```
sudo systemctl restart php-fpm
sudo systemctl restart nginx
```

Morto Ian Murdock, il padre di Debian

Il 28 dicembre 2015 l'informatica perde un illustre personaggio, il fondatore della distribuzione GNU/Linux Debian Ian Murdock.

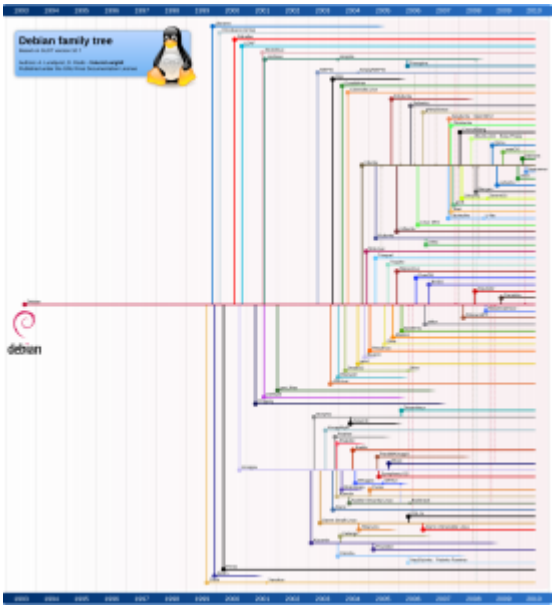
Si proprio lei, la Debian, il nome poetico la contraddistingue dalle altre: Deb da Debra, sua ragazza nel 1993, e Ian dal suo nome.
Ian Murdock pubblica "The Debian Manifesto", la filosofia della nuova distribuzione, la Debian 0.91, apertura dello sviluppo a tutta la comunità informatica, collaborazione con la *Free Software Foundation* e, cosa più importante, creare una distribuzione solida, ben mantenuta e che non diventi mai un prodotto commerciale. Partendo da queste basi Debian darà vita a molte altre distribuzioni diventando "*The universal operating system*".

Ian Murdock con Debian ha anche sviluppato l'**Advanced Packaging Tool**, conosciuto con l'acronimo **APT**, il gestore standard di pacchetti software. Una curiosità di APT è il print a video del suo help: con il comando "apt-get help", al termine della lista dei comandi e opzioni da passare all'APT, viene mostrata la scritta "This APT has Super Cow Powers".

Daniel Burrows nel '99 implementa "aptitude" inserendo il suo Easter Egg "does not have Super Cow Powers" e un riferimento al "Piccolo Principe".

http://dtricarico.photogulp.net/2009/03/super-mucca-debian-cowsay-fortune.html

La sua *distro* è considerata una delle più pure e aderenti ai principi ispiratori del software libero; nel 1996 Murdock divenne **CTO (*Chief Technology Officer*) della Linux Foundation**, per poi passare a Sun nel 2003



con il ruolo di Vice Presidente per le piattaforme emergenti. Qui il suo lavoro contribuì alla nascita di OpenSolaris, sistema che fu abbandonato quando Sun Microsystems fu acquisita da Oracle (27 gennaio 2010), nello stesso momento Murdock lasciò la società.

La sua morte lascia un'aria di mistero per via di un arresto violento la sera di sabato 26 dicembre 2015.

SFBAY.CA ha pubblicato un resoconto degli eventi:

http://sfbay.ca/2015/12/31/police-confirm-ian-murdock-arrest-before-suicide/

> He didn't indicate at any point in the jail booking process that he was suicidal and was medically examined again in jail, she said.
>
> On Monday, police returned to the 2400 block of Green Street on reports of a possible suicide. The city medical examiner's office confirmed Murdock was found dead there.

Lunedì scorso Murdock ha scritto online un messaggio che sembrava indicare un intento suicida (*"I'm committing suicide tonight…do not intervene as I have many stories to tell and do not want them to die with me #debian #runnerkrysty67"*).

La comunità ha pubblicato le istruzioni per porgere le condoglianze al seguente link:

https://bits.debian.org/2015/12/mourning-ian-murdock.html

> La sua famiglia in questo momento difficile ha chiesto di rispettare la loro privacy e noi vogliamo onorare questa loro richiesta.
>
> All'interno della nostra Debian e della più grande comunità Linux

> le condoglianze possono essere inviate a in-memoriam-ian@debian.org in modo da poterle archiviare e conservare.

E-Mail con virus Zero Day

Ho preso un virus ma non ho fatto nulla!

Esclamazione che ho sentito molto spesso nel corso di un 2015 dove gli hacker e i cracker cercano sempre più spesso di trarre profitto dall'ingenuità altrui.

Il metodo più facile per infettare un PC è usare uno ZERO DAY, un virus che nessun antivirus può intercettare; ma come farlo arrivare a destinazione? Mi aspetto che i sistemi antispam riconoscano i mittenti malintenzionati e come tali ne scartino le mail. Bene, allora sfruttiamo i mail server "puliti" per recapitare delle email valide.

Iniziamo con il capire cosa è uno 0-DAY, wikipedia ci è di aiuto con una spiegazione semplice:

> In informatica si definisce **0-day** qualsiasi vulnerabilità non nota e, per estensione, indica un tipo di attacco informatico che inizia nel "giorno zero", cioè nel momento in cui viene scoperta una falla di sicurezza in un sistema informatico. Questo tipo di attacco può mietere molte vittime proprio perché è lanciato quando ancora non è stata distribuita alcuna patch, e quindi i sistemi non sono ancora protetti.
>
> Normalmente si parla di 0-day (o zero-day) riferendosi ad essi come attività espressamente dolose compiute da cracker che si adoperano per trovarle proprio con l'intenzione di guadagnarsi un accesso abusivo ad un sistema informatico vulnerabile.
>
> ….
>
> Gli 0-day sono tra i peggiori pericoli del web, in quanto sono noti solo a una ristretta cerchia di cracker, e possono causare numerosi danni prima di essere scoperti.

Come funziona un antivirus? I virus noti vengono censiti nelle banche dati delle aziende produttrici di software antivirus, i programmi installati sui

nostri PC (o server) devono aggiornarsi costantemente al fine di avere le "signature" allineate con la banca dati centrale, questo per evitare virus messi in circolazione dalla data di ultimo aggiornamento possano infettare il sistema.

Ecco, quindi la cosa più importante, non è cosa fate con i vostri sistemi, ma quanto viene eseguito l'update delle signature.

Partendo da questo principio è logico pensare che le aziende produttrici aggiornino il database dei virus ad ogni nuova scoperta e/o segnalazione di software malevolo, bisogna poi attendere che questa informazione arrivi anche al nostro sistema.

Abbiamo quindi un lasso di tempo più o meno breve, ma utile per infettare milioni di dispositivi in giro per il mondo, se agiamo in fretta.

Preparato un virus anche banale non ci resta che metterlo in giro nel minor tempo possibile, sperando che gli utenti eseguano il software pericoloso, per fare questo dobbiamo aver preventivamente avuto accesso (manco a dirlo in modo illegale) a svariati server afflitti da vulnerabilità note per poterli usare come MAILSERVER e distribuire il nostro pacchetto.

Ed eccoci, tutto pronto, prepariamo la mail ad opera d'arte, con una buona traduzione in base alla nazione target, una contraffazione di un'azienda nota per rendere tutto meno sospetto e via, distribuzione in corso.

Ecco perché sempre più spesso vediamo nella nostra casella di posta mail che non aspettavamo, con fatture o preventivi provenienti da aziende note.

Se abbiamo dubbi sul file che ci è stato inviato, visto che il nostro antivirus non lo ha segnalato come malevolo, possiamo usare il servizio che mette a disposizione gratuitamente Virus Total:

https://www.virustotal.com/

Questo ci consente di verificare se e quale antivirus riconosce file sospetto come potenziale virus.

Anche se usiamo servizi come MAILPROTECTION facciamo attenzione ad aprire gli allegati, questo perché nonostante gli antivirus siano aggiornati con un'alta frequenza, ci sarà sempre una fascia oraria in cui ne potrà passare uno.

Vulnerabilità WordPress SEO by Yoast

Vulnerabilità SEO, ma come risolvere i danni fatti?

Come scritto nel blog ufficiale (LINK) il blasonato plugin SEO by Yoast

soffre di una gravissima vulnerabilità: **Blind SQL Injection,** file interessato sarebbe il ***class-bulk-editor-list-table.php***.

Cos'è una **Blind SQL Injection** e come possiamo sfruttarla?

Un hacker inserisce una query SQL non valida in un'applicazione, nel nostro caso **WordPress** che avendo un autore, un admin o un editor già autenticati che visitano un URL malformato, il malintenzionato riesce ad accedere e modificare il database **WordPress.**

Vediamo cosa è successo proprio a noi che scriviamo questo articolo.

Il furbo di turno ha sfruttato la vulnerabilità per modificare il database, creare un nuovo utente, concedergli i privilegi amministrativi ed aggiungere delle widget con codice javascript.

Tale codice servire a modificare i link del sito per rimandare a pubblicità, il modo più veloce per monetizzare. Fortunatamente l'hacker aveva un suo scopo ben preciso, quello di monetizzare, per questo motivo non ha fatto danni.

Questo serve a riflettere su quanti usano WordPress per scopi professionali senza affidarsi ad aziende o professionisti del settore.

Come suggerito da hostingtalk.it:

> In casi come questi, il consiglio è di **aggiornare immediatamente** il plugin WordPress SEO by Yoast all'ultima versione disponibile o di **affidarsi a servizi di hosting gestiti**, che eseguono in automatico per l'utenza gli upgrade di sicurezza necessaria. Altra alternativa è **l'autoaggiornamento di WordPress**, sempre che non sia stato disabilitato.

In alternativa un contratto di manutenzione può salvare il proprio business.

Abilitare diverse versioni di PHP in PLESK

Amministrando un web server con PLESK prima o poi arriva la richiesta di installare una seconda versione di PHP e di renderla disponibile ai clienti attraverso il pannello PLESK.

In questa guida l'installazione è stata fatta su una macchina CentOS 6.6 e

Plesk 11.5.30:

Per prima cosa creiamo la directory dove potere scaricare il pacchetto PHP

```
cd /usr/local/src
# mkdir php562
# cd php562
wget http://it1.php.net/get/php-5.6.2.tar.gz/from/this/mirror
mv mirror php-5.6.2.tar.gz
tar -xvzf php-5.6.2.tar.gz
cd php-5.6.2
```

Siamo pronti per iniziare, configuriamo per la compilazione:

```
./configure '--with-libdir=lib64' '--cache-file=../config.cache' '--
prefix=/usr/local/php562-cgi' '--with-config-file-path=/usr/local/php562-
cgi/etc' '--disable-debug' '--with-pic' '--disable-rpath' '--enable-fastcgi'
'--with-bz2' '--with-curl' '--with-xpm-dir=/usr/local/php562-cgi' '--with-
png-dir=/usr/local/php562-cgi' '--enable-gd-native-ttf' '--without-gdbm' '--
with-gettext' '--with-gmp' '--with-iconv' '--with-jpeg-dir=/usr/local/php562-
cgi' '--with-openssl' '--with-pspell' '--with-pcre-regex' '--with-zlib' '--
enable-exif' '--enable-ftp' '--enable-sockets' '--enable-sysvsem' '--enable-
sysvshm' '--enable-sysvmsg' '--enable-wddx' '--with-kerberos' '--with-
unixODBC=/usr' '--enable-shmop' '--enable-calendar' '--without-sqlite3' '--
with-libxml-dir=/usr/local/php562-cgi' '--enable-pcntl' '--with-imap' '--
with-imap-ssl' '--enable-mbstring' '--enable-mbregex' '--with-gd' '--enable-
bcmath' '--with-xmlrpc' '--with-ldap' '--with-ldap-sasl' '--with-mysql=/usr'
'--with-mysqli' '--with-snmp' '--enable-soap' '--with-xsl' '--enable-
xmlreader' '--enable-xmlwriter' '--enable-pdo' '--with-pdo-mysql' '--with-
pdo-pgsql' '--with-pear=/usr/local/php562-cgi/pear' '--with-mcrypt' '--
enable-intl' '--without-pdo-sqlite' '--with-config-file-scan-
dir=/usr/local/php562-cgi/php.d' --enable-shared --enable-zip
```

Ora il classico make e poi make install, mi raccomando non lanciate make test

```
make
make install
```

Copiamo il php.ini sotto nella directory php562-cgi

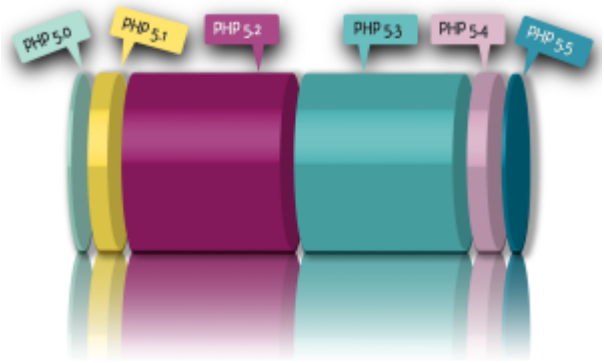
```
cp php.ini-development /usr/local/php562-cgi/php.ini
```

Ora non rimane che censire nel pannello PLESK la nuova versione di PHP

```
/usr/local/psa/bin/php_handler --add -displayname 5.6.2 -path
/usr/local/php562-cgi/bin/php-cgi -phpini /usr/local/php562-cgi/php.ini -type
fastcgi -id 5.6.2
```

Nel mio caso la prima installazione di PHP non è stata così liscia, ho dovuto installare alcuni pacchetti:

```
yum install bzip2-devel.x86_64 bzip2.x86_64
yum install libjpeg*
yum install libpng-devel
yum install freetype
yum install libXpm-devel
yum install libgmp3-dev gmp.x86_64 gmp-devel.x86_64
yum install openssl openssl-devel pam-devel
yum install pam-devel
yum install libicu-devel libc-client-devel.x86_64 libc-client.x86_64
yum install libtomcrypt-devel.x86_64 libmcrypt-devel.x86_64 php-mcrypt.x86_64
yum install unixODBC-devel
yum install postgresql-devel postgresql-libs
yum install pspell php-pspell.x86_64 aspell-devel net-snmp-devel libxslt-
devel libxml2-devel pcre-devel t1lib-devel.x86_64 libtidy-devel php-pecl-zip
```



Size of MySQL database

Vogliamo sapere lo spazio occupato da ogni singolo database usando la command line, una semplice query restituisce a video l'informazione richiesta.

Prestate attenzione perché questa query potrebbe richiedere molto tempo per DB di grandi dimensioni.

```
mysql> SELECT table_schema "DB Name",
Round(Sum(data_length + index_length) / 1024 / 1024, 1) "DB Size in MB"
FROM information_schema.tables
GROUP BY table_schema;
```

Ecco un esempio dell'output:

```
+--------------------------------+---------------+
| DB Name | DB Size in MB |
+--------------------------------+---------------+
| monitoraggio | 1505.0 |
| mysql | 0.7 |
| pcparts | 0.4 |
```

```
| performance_schema | 0.0 |
| photogulp | 193.3 |
| photogulp_webalbum | 0.5 |
| phplistdb | 33.4 |
| pixellone_artistika | 7.9 |
| pixellone_enter | 0.4 |
| rc-bazar_oc | 4.8 |
| wordpress_9 | 1.1 |
+---------------------------------+--------------+
```

OpenVPN gateway internet [CentOS 6.6]

Usare OpenVPN per accedere ad un'infrastruttura e uscire su internet direttamente dal server VPN.

Lo scenario è quello di avere dei consulenti in giro per clienti che si collegano ad internet per mezzo del proxy del cliente, questo blocca le connessioni di tutti i client, a partire da quello di posta (Outlook, Thunderbird, Mail, ecc…)

Iniziamo con la configurazione del server.

L'articolo tratta l'installazione del software su un sistema operativo Debian Squeeze, ma a pacchetti installati, le informazioni sono utilizzabili sulle più diffuse distribuzioni.

Diamo per scontato che la porta 443 TCP verso il vostro server sia raggiungibile.

Il primo step è naturalmente quello di installare openvpn:

```
# yum install openvpn.x86_64
```

Generazione dei certificati

Il pacchetto di OpenVPN fornisce una serie di script già pronti atti a tale scopo nel path */usr/share/doc/openvpn-2.2.2/easy-rsa/2.0/*:

```
# ls /usr/share/doc/openvpn-2.2.2/easy-rsa/2.0/
build-ca     build-key-pass    build-req-pass  Makefile          pkitool      vars
build-dh     build-key-pkcs12  clean-all       openssl-0.9.6.cnf README       whichopensslcnf
build-inter  build-key-server  inherit-inter   openssl-0.9.8.cnf revoke-full
build-key    build-req         list-crl        openssl-1.0.0.cnf sign-req
```

Per comodità spostiamo tutta la directory sotto */etc/openvpn/rsa/*.

```
# cp -r /usr/share/doc/openvpn-2.2.2/easy-rsa/2.0/ /etc/openvpn/rsa
# cd /etc/openvpn/rsa
```

Apriamo il file "*vars*" e editiamo i campi, questo velocizzerà la creazione dei certificato, è comodo per chi ha la necessità di creare molti certificati.

I parametri da modificare sono i seguenti:

- KEY_SIZE
- KEY_COUNTRY
- KEY_PROVINCE
- KEY_CITY
- KEY_ORG
- KEY_EMAIL

Un esempio del file vars:

```
export KEY_SIZE=1024
...
export KEY_COUNTRY="IT"
export KEY_PROVINCE="IT"
export KEY_CITY="Roma"
export KEY_ORG="LBIT"
export KEY_EMAIL="vpn@lbit-solution.it"
```

A questo punto siamo pronti per generare la nostra **CA (certificate authority)**

```
# chmod 755 /etc/openvpn/rsa/whichopensslcnf
# chmod 755 clean-all
# source vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on
/etc/openvpn/rsa/keys
# ./clean-all
```

È necessario richiamare anche lo script "clean-all" per iniziare con un ambiente pulito.

Ora possiamo generare la nostra **Certificate Authority**:

```
# chmod 755 build-ca
# chmod 755 /etc/openvpn/rsa/pkitool
# ./build-ca
Generating a 1024 bit RSA private key
.....................................................++++++
...++++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [IT]:
State or Province Name (full name) [IT]:
Locality Name (eg, city) [Roma]:
Organization Name (eg, company) [LBIT]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [LBIT CA]:
Email Address [vpn@lbit-solution.it]:
```

Avendo preconfigurato il file “vars” è sufficiente premere invio visto che il sistema ci propone come default i valori che avevamo inserito ad inizio procedura.
Ora possiamo creare il certificato per il server VPN:

```
# ./build-key-server GatewayVPN
```

GatewayVPN è il nome della macchina su cui sto installando il server VPN, per coerenza la coppia chiave/certificato avrà il nome dell’host su cui viene usato.

Per evitare che ad ogni riavvio di OpenVPN sia richiesta una password premere invio senza inserire nulla alla richiesta di password:

```
Generating a 1024 bit RSA private key
..................++++++
.++++++
writing new private key to 'GatewayVPN.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [IT]:
State or Province Name (full name) [IT]:
Locality Name (eg, city) [Roma]:
Organization Name (eg, company) [LBIT]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [GatewayVPN]:
Email Address [vpn@lbit-solution.it]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:password
An optional company name []:
Using configuration from /etc/openvpn/rsa/openssl.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'IT'
stateOrProvinceName   :PRINTABLE:'IT'
localityName          :PRINTABLE:'Roma'
organizationName      :PRINTABLE:'LBIT'
commonName            :PRINTABLE:'GatewayVPN'
emailAddress          :IA5STRING:'vpn@lbit-solution.it'
Certificate is to be certified until Apr 25 13:50:00 2020 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Generiamo ora il file Diffie-Hellman, necessario per l’avvio delle connessioni cifrate.

```
# chmod 755 build-dh
# ./build-dh
Generating DH parameters, 1024 bit long safe prime, generator 2
This is going to take a long time
........+.............
```

Generiamo l’ultima chiave necessaria per l’instaurazione di una connessione sicura

```
# openvpn --genkey --secret keys/ta.key
```

Generazione dei certificati per i client

La procedura per generare i certificati dei client è identica a quella del server, nell’esempio li creiamo nominali per una semplice identificazione, in caso di grandi numeri è possible usare la matricola aziendale.

```
# chmod 755 build-key
#  ./build-key mcapasso
  Please edit the vars script to reflect your configuration,
  then source it with "source ./vars".
  Next, to start with a fresh PKI configuration and to delete any
  previous certificates and keys, run "./clean-all".
  Finally, you can run this tool (pkitool) to build certificates/keys.
root@webdav:/etc/openvpn/easy-rsa# source ./vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on /etc/openvpn/easy-rsa/keys
root@webdav:/etc/openvpn/easy-rsa#  ./build-key mcapasso
Generating a 1024 bit RSA private key
.....................................++++++
.............................................++++++
writing new private key to 'mcapasso.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [IT]:
State or Province Name (full name) [RM]:
Locality Name (eg, city) [Roma]:
Organization Name (eg, company) [LBIT]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [mcapasso]:
Name []:Mirko Capasso
Email Address [supporto@lbit-solution.it]:mcapasso@lbit-solution.it

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /etc/openvpn/easy-rsa/openssl.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'IT'
stateOrProvinceName   :PRINTABLE:'RM'
localityName          :PRINTABLE:'Roma'
organizationName      :PRINTABLE:'LBIT'
commonName            :PRINTABLE:'mcapasso'
name                  :PRINTABLE:'Mirko Capasso'
emailAddress          :IA5STRING:'mcapasso@lbit-solution.it'
Certificate is to be certified until Oct 19 14:29:37 2024 GMT (3650 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Configurazione del server

Ora andiamo a configurare il demone OpenVPN, anche in questo caso il pacchetto dovrebbe portare con se degli esempi.

```
# cp /usr/share/doc/openvpn-2.2.2/sample-config-files/server.conf
/etc/openvpn/
```

Di seguito un file di configurazione, dopo andiamo a spiegare le direttive:

```
# SERVER CONF
port 443
proto tcp
dev tun

ca rsa/keys/ca.crt
cert rsa/keys/GatewayVPN.crt
key rsa/keys/GatewayVPN.key
dh rsa/keys/dh1024.pem

client-config-dir ccd
server 10.1.1.0  255.255.255.0
route 10.1.1.0 255.255.255.0
ifconfig-pool-persist ipp.txt
cipher AES-256-CBC
comp-lzo
persist-key
persist-tun

status /var/log/openvpn-status.log 5
status-version 2
log-append /var/log/openvpn-status.log
verb 3  # verbose mode

# ROUTE THE CLIENT'S INTERNET ACCESS THROUGH THIS SERVER:
push "redirect-gateway def1"
push "remote-gateway 10.1.1.1"
push "dhcp-option DNS 8.8.8.8"
keepalive 10 60
```

La prima entry “*port*” è la porta sulla quale il servizio OpenVPN si metterà in ascolto, “*proto*” il protocollo, possiamo usare TCP o UDP, in questo scenario abbiamo scelto TCP per evitare che le connessioni UDP fossero droppate da firewall o proxy.

Non abbiamo usato la entry “*local*” poiché il nostro serve deve accettare connessioni su tutte le interfacce di rete, nel caso in cui ci fossero più

interfacce ma solo una destinata al demone allora sarà necessario indicare l'IP sul quale mettersi in ascolto, come l'esempio seguente:

```
local 10.10.256.25
```

Possiamo usare un tunnel al layer 3 del livello OSI, (**tap**) oppure un bridge di rete a livello 2 (**tun**), nel nostro file abbiamo inserito la seconda opzione.

A seguire la parte relativa ai certificati:

```
ca rsa/keys/ca.crt
cert rsa/keys/GatewayVPN.crt
key rsa/keys/GatewayVPN.key
dh rsa/keys/dh1024.pem
```

Le direttive da non dimenticare per consentire l'accesso ad internet tramite VPN sono le ultime, al posto di 10.1.1.1 va inserito l'IP della scheda tun0:

```
# ROUTE THE CLIENT'S INTERNET ACCESS THROUGH THIS SERVER:
push "redirect-gateway def1"
push "remote-gateway 10.1.1.1"
push "dhcp-option DNS 8.8.8.8"
```

Configurazione di IPTABLES

Per consentire ai client di uscire su internet tramite il gateway VPN andiamo ad abilitare il fowarding e il MASQUERADE tramite IPTABLES:

```
sysctl -w net.ipv4.ip_forward=1
iptables -t nat -A POSTROUTING -s 10.1.1.0/24 -o eth0 -j MASQUERADE
```

Se abbiamo IPTABLES configurato andiamo ad aggiungere anche le policy di ACCEPT:

```
iptables -A INPUT -i tun0 -j ACCEPT
iptables -A FORWARD -i tun0 -j ACCEPT
```

Per impostare in modo permanente le regole IPTABLES sopra descritte editiamo il file /etc/sysconfig/iptables:

```
# vi /etc/sysconfig/iptables

# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -o eth0 -j MASQUERADE
COMMIT

*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -i tun0 -j ACCEPT
-A FORWARD -i tun0 -o eth0 -j ACCEPT
-A FORWARD -i eth0 -o tun0 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

Avviare il demone di OpenVPN e configurare i certificati dei client.

Configurazione dei client

Per prima cosa dobbiamo copiarci i certificati:

- La coppia certificato/chiave per il client (i due file .key e .crt)
- Il certificato della CA del server (il file ca.crt)
- La chiave di autenticazione TLS (il file ta.key)

Il file di configurazione di una macchina Windows non è complicato ma al primo errore smette di funzionare senza scrivere nei log:

```
client
dev tun
proto tcp
remote IP_SERVER_VPN 443
resolv-retry infinite
nobind
persist-key
persist-tun
# THE CSR FILE:
ca "C:\\Program Files\\OpenVPN\\config\\ca.crt"
cert "C:\\Program Files\\OpenVPN\\config\\dtricarico.crt"
key "C:\\Program Files\\OpenVPN\\config\\dtricarico.key"
ns-cert-type server
cipher AES-256-CBC
comp-lzo
redirect-gateway def1
verb 3
route-method exe
route-delay 2
```

OpenVPN gateway internet [Debian]

Usare OpenVPN per accedere ad un'infrastruttura e uscire su internet direttamente dal server VPN.

Lo scenario è quello di avere dei consulenti in giro per clienti che si collegano ad internet per mezzo del proxy del cliente, questo blocca le connessioni di tutti i client, a partire da quello di posta (Outlook, Thunderbird, Mail, ecc…)

Iniziamo con la configurazione del server.

L'articolo tratta l'installazione del software su un sistema operativo Debian Squeeze, ma a pacchetti installati, le informazioni sono utilizzabili sulle più diffuse distribuzioni.

Diamo per scontato che la porta 443 TCP verso il vostro server sia raggiungibile.

Il primo step è naturalmente quello di installare openvpn:

```
# apt-get install openvpn
```

Generazione dei certificati

Il pacchetto di OpenVPN fornisce una serie di script già pronti atti a tale scopo nel path */usr/share/doc/openvpn/examples/easy-rsa/2.0/*:

```
# ls /usr/share/doc/openvpn/examples/easy-rsa/2.0/
build-ca          build-key-server  Makefile              sign-req
build-dh          build-req         openssl-0.9.6.cnf.gz  vars
build-inter       build-req-pass    openssl.cnf           whichopensslcnf
build-key         clean-all         pkitool
build-key-pass    inherit-inter     README.gz
build-key-pkcs12  list-crl          revoke-full
```

Per comodità spostiamo tutta la directory sotto */etc/openvpn/rsa/*.

```
# cp -r /usr/share/doc/openvpn/examples/easy-rsa/2.0/ /etc/openvpn/rsa
# cd /etc/openvpn/rsa
```

Apriamo il file "*vars*" e editiamo i campi, questo velocizzerà la creazione

dei certificato, è comodo per chi ha la necessità di creare molti certificati.

I parametri da modificare sono i seguenti:

- KEY_SIZE
- KEY_COUNTRY
- KEY_PROVINCE
- KEY_CITY
- KEY_ORG
- KEY_EMAIL

Un esempio del file vars:

```
export KEY_SIZE=1024
...
export KEY_COUNTRY="IT"
export KEY_PROVINCE="IT"
export KEY_CITY="Roma"
export KEY_ORG="LBIT"
export KEY_EMAIL="vpn@lbit-solution.it"
```

A questo punto siamo pronti per generare la nostra **CA (certificate authority)**

```
# source vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on
/etc/openvpn/rsa/keys
# ./clean-all
```

È necessario richiamare anche lo script “clean-all” per iniziare con un ambiente pulito.

Ora possiamo generare la nostra **Certificate Authority**:

```
# ./build-ca
Generating a 1024 bit RSA private key
.....................................................++++++
...++++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [IT]:
State or Province Name (full name) [IT]:
Locality Name (eg, city) [Roma]:
Organization Name (eg, company) [LBIT]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [LBIT CA]:
Email Address [vpn@lbit-solution.it]:
```

Avendo preconfigurato il file “vars” è sufficiente premere invio visto che il sistema ci propone come default i valori che avevamo inserito ad inizio procedura.
Ora possiamo creare il certificato per il server VPN:

```
# ./build-key-server GatewayVPN
```

GatewayVPN è il nome della macchina su cui sto installando il server VPN, per coerenza la coppia chiave/certificato avrà il nome dell’host su cui viene usato.

Per evitare che ad ogni riavvio di OpenVPN sia richiesta una password premere invio senza inserire nulla alla richiesta di password:

```
Generating a 1024 bit RSA private key
..................++++++
.++++++
writing new private key to 'GatewayVPN.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [IT]:
State or Province Name (full name) [IT]:
Locality Name (eg, city) [Roma]:
Organization Name (eg, company) [LBIT]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [GatewayVPN]:
Email Address [vpn@lbit-solution.it]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:password
An optional company name []:
Using configuration from /etc/openvpn/rsa/openssl.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'IT'
stateOrProvinceName   :PRINTABLE:'IT'
localityName          :PRINTABLE:'Roma'
organizationName      :PRINTABLE:'LBIT'
commonName            :PRINTABLE:'GatewayVPN'
emailAddress          :IA5STRING:'vpn@lbit-solution.it'
Certificate is to be certified until Apr 25 13:50:00 2020 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Generiamo ora il file Diffie-Hellman, necessario per l’avvio delle connessioni cifrate.

```
# ./build-dh
Generating DH parameters, 1024 bit long safe prime, generator 2
This is going to take a long time
........+.............
```

Generiamo l’ultima chiave necessaria per l’instaurazione di una connessione sicura

```
# openvpn --genkey --secret keys/ta.key
```

Generazione dei certificati per i client

La procedura per generare i certificati dei client è identica a quella del server, nell’esempio li creiamo nominali per una semplice identificazione, in caso di grandi numeri è possible usare la matricola aziendale.

```
#  ./build-key mcapasso
  Please edit the vars script to reflect your configuration,
  then source it with "source ./vars".
  Next, to start with a fresh PKI configuration and to delete any
  previous certificates and keys, run "./clean-all".
  Finally, you can run this tool (pkitool) to build certificates/keys.
root@webdav:/etc/openvpn/easy-rsa# source ./vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on /etc/openvpn/easy-
rsa/keys
root@webdav:/etc/openvpn/easy-rsa#  ./build-key mcapasso
Generating a 1024 bit RSA private key
.....................................++++++
.....................................++++++
writing new private key to 'mcapasso.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [IT]:
State or Province Name (full name) [RM]:
Locality Name (eg, city) [Roma]:
Organization Name (eg, company) [LBIT]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [mcapasso]:
Name []:Mirko Capasso
Email Address [supporto@lbit-solution.it]:mcapasso@lbit-solution.it

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /etc/openvpn/easy-rsa/openssl.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'IT'
stateOrProvinceName   :PRINTABLE:'RM'
localityName          :PRINTABLE:'Roma'
organizationName      :PRINTABLE:'LBIT'
commonName            :PRINTABLE:'mcapasso'
name                  :PRINTABLE:'Mirko Capasso'
emailAddress          :IA5STRING:'mcapasso@lbit-solution.it'
Certificate is to be certified until Oct 19 14:29:37 2024 GMT (3650 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Configurazione del server

Ora andiamo a configurare il demone OpenVPN, anche in questo caso il pacchetto dovrebbe portare con se degli esempi.

```
# cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz
/etc/openvpn/
# cd /etc/openvpn
# gunzip server.conf.gz
```

Di seguito un file di configurazione, dopo andiamo a spiegare le direttive:

```
# SERVER CONF
port 443
proto tcp
dev tun

ca rsa/keys/ca.crt
cert rsa/keys/GatewayVPN.crt
key rsa/keys/GatewayVPN.key
dh rsa/keys/dh1024.pem

client-config-dir ccd
server 10.1.1.0  255.255.255.0
route 10.1.1.0 255.255.255.0
ifconfig-pool-persist ipp.txt
cipher AES-256-CBC
comp-lzo
persist-key
persist-tun

status /var/log/openvpn-status.log 5
status-version 2
log-append /var/log/openvpn-status.log
verb 3  # verbose mode

# ROUTE THE CLIENT'S INTERNET ACCESS THROUGH THIS SERVER:
push "redirect-gateway def1"
push "remote-gateway 10.1.1.1"
push "dhcp-option DNS 8.8.8.8"
keepalive 10 60
```

La prima entry “*port*” è la porta sulla quale il servizio OpenVPN si metterà in ascolto, “*proto*” il protocollo, possiamo usare TCP o UDP, in questo scenario abbiamo scelto TCP per evitare che le connessioni UDP fossero droppate da firewall o proxy.

Non abbiamo usato la entry “*local*” poiché il nostro serve deve accettare connessioni su tutte le interfacce di rete, nel caso in cui ci fossero più

interfacce ma solo una destinata al demone allora sarà necessario indicare l'IP sul quale mettersi in ascolto, come l'esempio seguente:

```
local 10.10.256.25
```

Possiamo usare un tunnel al layer 3 del livello OSI, (**tap**) oppure un bridge di rete a livello 2 (**tun**), nel nostro file abbiamo inserito la seconda opzione.

A seguire la parte relativa ai certificati:

```
ca rsa/keys/ca.crt
cert rsa/keys/GatewayVPN.crt
key rsa/keys/GatewayVPN.key
dh rsa/keys/dh1024.pem
```

Le direttive da non dimenticare per consentire l'accesso ad internet tramite VPN sono le ultime, al posto di 10.1.1.1 va inserito l'IP della scheda tun0:

```
# ROUTE THE CLIENT'S INTERNET ACCESS THROUGH THIS SERVER:
push "redirect-gateway def1"
push "remote-gateway 10.1.1.1"
push "dhcp-option DNS 8.8.8.8"
```

Configurazione di IPTABLES

Per consentire ai client di uscire su internet tramite il gateway VPN andiamo ad abilitare il fowarding e il MASQUERADE tramite IPTABLES:

```
sysctl -w net.ipv4.ip_forward=1
iptables -t nat -A POSTROUTING -s 10.1.1.0/24 -o eth0 -j MASQUERADE
```

Se abbiamo IPTABLES configurato andiamo ad aggiungere anche le policy di ACCEPT:

```
iptables -A INPUT -i tun0 -j ACCEPT
iptables -A FORWARD -i tun0 -j ACCEPT
```

Avviare il demone di OpenVPN e configurare i certificati dei client.

Configurazione dei client

Per prima cosa dobbiamo copiarci i certificati:

- La coppia certificato/chiave per il client (i due file .key e .crt)
- Il certificato della CA del server (il file ca.crt)

- La chiave di autenticazione TLS (il file ta.key)

Il file di configurazione di una macchina Windows non è complicato ma al primo errore smette di funzionare senza scrivere nei log:

```
client
dev tun
proto tcp
remote IP_SERVER_VPN 443
resolv-retry infinite
nobind
persist-key
persist-tun
# THE CSR FILE:
ca "C:\\Program Files\\OpenVPN\\config\\ca.crt"
cert "C:\\Program Files\\OpenVPN\\config\\dtricarico.crt"
key "C:\\Program Files\\OpenVPN\\config\\dtricarico.key"
ns-cert-type server
cipher AES-256-CBC
comp-lzo
redirect-gateway def1
verb 3
route-method exe
route-delay 2
```

Postfix Forward Email To Multiple Email Account

Inoltrare le email a più indirizzi di posta elettronica.

Dobbiamo abilitare i virtual domain nella configurazione di Postfix, editiamo il file *main.cf*

```
# vi /etc/postfix/main.cf
```

Andiamo ad inserire il nome a dominio per il quale vogliamo creare i virtual alias o i nomi a dominio se sono più siti e il path con gli alias.

```
virtual_alias_domains = 3load.com
virtual_alias_maps = hash:/etc/postfix/virtual
# virtual_alias_domains = 3load.com lbit-solution.it ...
```

Apriamo il file virtual

```
# vi /etc/postfix/virtual
```

Ora possiamo configurare ogni singola casella oppure ogni dominio, nel primo esempio inoltriamo tutte le mail di info a una casella gmail mentre le mail di supporto a più caselle di posta

```
info@3load.com   3load@gmail.com
supporto@3load.com   mailexample1@gmail.com  mailexample2@libero.it
```

Aggiungiamo anche la redirezione dell'intero dominio lbit-solution.it

```
info@3load.com   3load@gmail.com
supporto@3load.com   mailexample1@gmail.com  mailexample2@libero.it
@lbit-solution.it  mailexample@dominio.it
```

Salviamo il file ed eseguiamo il reload di postfix

```
# postmap /etc/postfix/virtual
# service postfix reload
```

Floating Widget Social Network

Sito pulito o widget dei social network?

Non esiste sito internet senza widget social o connessioni ai propri profili o pagine aziendali, il cliente vuole html pulito ma senza rinunciare al badge di google plus piuttosto che al "mi piace" di facebook.

La soluzione è metterli a scomparsa in modo da lasciare il sito pulito senza widget fastidiose e pacchiane.

Guarda la demo http://blog.lbit-solution.it//prova.html

Scarica il file di prova al seguente link:
http://blog.lbit-solution.it/wp-content/uploads/2014/04/prova.html.gz

La personalizzazione del codice è molto semplice, per facebook e google plus basta sostituire il nome account e l'ID del profilo:

likebox.php?href=http%3A%2F%2Ffacebook.com%2FLbitSoluzioniInformatiche&width

Al posto si "LbitSoluzioniInformatiche" inserisci l'account della pagina facebbok, per quanto riguarda google plus sostituisci l'ID del profilo:

g-page" data-width="240" data-
href="https://plus.google.com/109087407088989811737" data-rel="publisher">

in questo caso il mio ID è "109087407088989811737", è sufficiente inserire il proprio per avere il badge configurato.

Twitter invece richiede la creazione di una nuova widget direttamente dal sito twitter.com, nel codice che hai scaricato sostituisci tutto questo:

<!-- Start Twitter Badge -->

Tweets di @LinuxLBiT

<script>!function(d,s,id){var
js,fjs=d.getElementsByTagName(s)[0],p=/^http:/.test(d.location)?'http':'https
';if(!d.getElementById(id)){js=d.createElement(s);js.id=id;js.src=p+"://platf
orm.twitter.com/widgets.js";fjs.parentNode.insertBefore(js,fjs);}}(document,"
script","twitter-wjs");</script>

<!– // END Twitter Badge –>

Pochi passi per avere l'effetto desiderato.