Installation

insfrastructure, sécurité, vulnérabilités, scanner, Vuls, installation

Installation des paquets

Nous installons les paquets nécessaires

```
sudo aptitude install sqlite git gcc make wget g++
```

Installation des sources

Nous téléchargeons les sources de go

```
wget https://dl.google.com/go/go1.14.2.linux-amd64.tar.gz
```

Nous décompressons l'archive dans le répertoire /usr/local

```
sudo tar -C /usr/local -xzf go1.14.2.linux-amd64.tar.gz
```

Nous créons le répertoire de travail

```
mkdir $HOME/go
```

Nous créons le fichier de variables d'environnement /etc/profile.d/goenv.sh

```
sudo vi /etc/profile.d/goenv.sh
```

Nous y ajoutons ces variables

```
export GOROOT=/usr/local/go
export GOPATH=$HOME/go
export PATH=$PATH:$GOROOT/bin:$GOPATH/bin
```

Nous chargeons les variables

```
source /etc/profile.d/goenv.sh
```

Déploiement go-cve-dictionary

Pour compiler go-cve-dictionnary

```
$ sudo mkdir /var/log/vuls
$ sudo chown <user> /var/log/vuls
```

```
$ sudo chmod 700 /var/log/vuls
$ mkdir -p $GOPATH/src/github.com/vulsio
$ cd $GOPATH/src/github.com/vulsio
$ git clone https://github.com/vulsio/go-cve-dictionary.git
$ cd go-cve-dictionary
$ make install
```

Si vous avez une erreur de la compilation de sqlite, c'est qu'il vous manque le paquet g++. Pour l'installer

```
sudo aptitude install g++
```

Pour récupérer les data CVE de puis NVD

```
$ cd $HOME
$ for i in `seq 2002 $(date +"%Y")`; do go-cve-dictionary fetchnvd -years
$i; done
... snip ...
$ ls -alh cve.sqlite3
-rw-r--r--. 1 centos centos  51M Aug  6 08:10 cve.sqlite3
-rw-r--r--. 1 centos centos  32K Aug  6 08:10 cve.sqlite3-shm
-rw-r--r--. 1 centos centos 5.1M Aug  6 08:10 cve.sqlite3-wal
```

Déploiement goval-dictionary

Pour compiler goval-dictionary

```
$ mkdir -p $GOPATH/src/github.com/kotakanbe
$ cd $GOPATH/src/github.com/kotakanbe
$ git clone https://github.com/kotakanbe/goval-dictionary.git
$ cd goval-dictionary
$ make install
$ ln -s $GOPATH/src/github.com/kotakanbe/goval-dictionary/oval.sqlite3
$HOME/oval.sqlite3
```

Pour récupérer les data OVAL pour Debian Buster

```
goval-dictionary fetch-debian 10
```

Déploiement gost

Pour compiler gost

```
$ sudo mkdir /var/log/gost
$ sudo chown <user> /var/log/gost
$ sudo chmod 700 /var/log/gost
$ mkdir -p $GOPATH/src/github.com/knqyf263
```

```
$ cd $GOPATH/src/github.com/knqyf263
$ git clone https://github.com/knqyf263/gost.git
$ cd gost
$ make install
$ ln -s $GOPATH/src/github.com/knqyf263/gost/gost.sqlite3 $HOME/gost.sqlite3
```

Pour mettre à jour les définitions pour Debian

```
$ gost fetch debian
```

Déploiement go-exploitdb

Pour compiler go-exploitdb

```
$ sudo mkdir /var/log/go-exploitdb
$ sudo chown <user> /var/log/go-exploitdb
$ sudo chmod 700 /var/log/go-exploitdb
$ mkdir -p $GOPATH/src/github.com/mozqnet
$ cd $GOPATH/src/github.com/mozqnet
$ git clone https://github.com/mozqnet/go-exploitdb.git
$ cd go-exploitdb
$ make install
$ ln -s $GOPATH/src/github.com/mozqnet/go-exploitdb/go-exploitdb.sqlite3
$HOME/go-exploitdb.sqlite3
```

Pour mettre à jour la base des exploits

```
$ go-exploitdb fetch exploitdb
```

Déploiement go-msfdb

Pour compiler go-msfdb

```
$ sudo mkdir /var/log/go-msfdb
$ sudo chown <user> /var/log/go-msfdb
$ sudo chmod 700 /var/log/go-msfdb
$ mkdir -p $GOPATH/src/github.com/takuzoo3868
$ cd $GOPATH/src/github.com/takuzoo3868
$ git clone https://github.com/takuzoo3868/go-msfdb.git
$ cd go-msfdb
$ make install
$ ln -s $GOPATH/src/github.com/takuzoo3868/go-msfdb/go-msfdb.sqlite3
$HOME/go-msfdb.sqlite3
```

Pour mettre à jour la base Metasploit

```
$ go-msfdb fetch msfdb
```

Déploiement Vuls

Pour compiler Vuls

```
$ mkdir -p $GOPATH/src/github.com/future-architect
$ cd $GOPATH/src/github.com/future-architect
$ git clone https://github.com/future-architect/vuls.git
$ cd vuls
$ make install
```

Pour mettre à jour Vuls

```
$ rm -rf $GOPATH/pkg/linux_amd64/github.com/future-architect/vuls/
$ rm -rf $GOPATH/src/github.com/future-architect/vuls/
$ cd $GOPATH/src/github.com/future-architect
$ git clone https://github.com/future-architect/vuls.git
$ cd vuls
$ make install
```

Configuration des hosts

Nous créons le fichier de configuration des hosts

```
$ vi config.toml
```

Nous y ajoutons le serveur local

```
[servers]

[servers.localhost]
host = "localhost"
port = "local"
```

Liens

- installation manuelle
- installation de GO
- go-cve-dictionary
- goval-dictionary
- Go security tracker
- go-exploitdb
- go-msfdb

From:
https://wiki.grohub.org/ - **Grohub wiki**

Permanent link:
https://wiki.grohub.org/infrastructure/securite/vuls/installation

Last update: **01/01/2022 20:19**