L'audit de configuration avec pyCAF

Maxime OLIVIER

–

SSTIC le 5 juin 2015

Plan

- L'audit de configuration
- Les contraintes
- La réponse
- L'outil
- Les limitations
- Les évolutions

L'audit de configuration

- Conformité à un référentiel

L'audit de configuration

- Conformité à un référentiel
- Recherche de vulnérabilité

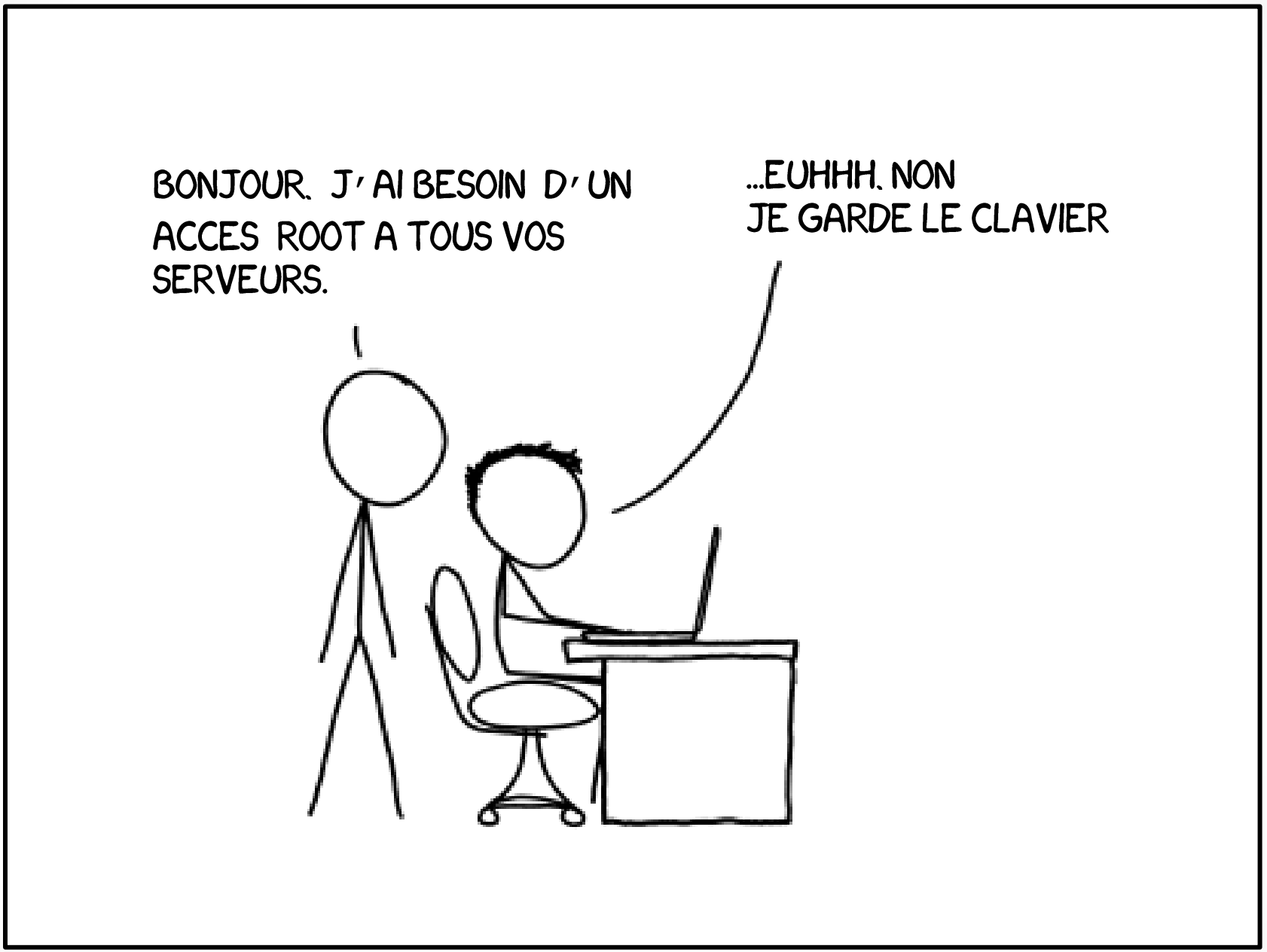
L'audit de configuration

- Conformité à un référentiel
- Recherche de vulnérabilité
- Valider les résultats d'un audit d'architecture

L'audit de configuration

- Conformité à un référentiel
- Recherche de vulnérabilité
- Valider les résultats d'un audit d'architecture
- Valider les résultats d'un audit de code source

BONJOUR. J'AI BESOIN D'UN ACCES ROOT A TOUS VOS SERVEURS.


BONJOUR. J'AI BESOIN D'UN ACCES ROOT A TOUS VOS SERVEURS.
...EUHHH. NON JE GARDE LE CLAVIER

D'ACCORD. ON VA PREVOIR CINQ JOURS ENSEMBLE POUR LES VERIFICATIONS.


D'ACCORD. ON VA PREVOIR CINQ JOURS ENSEMBLE POUR LES VERIFICATIONS.
J'AI 1H30 A VOUS ACCORDER

OK. TOUS VOS SERVEURS SONT HOMOGENES EN VERSION ET EN CONFIGURATION ?

OK. TOUS VOS SERVEURS SONT HOMOGENES EN VERSION ET EN CONFIGURATION ?

OUI... EUH. ON A UN DOMAINE AVEC LA PLUPART DES SERVEURS. QUELQUES SERVEURS REDHAT ET HISTORIQUEMENT DES DEBIAN QUI DOIVENT TRAINER...

La réponse

Par l'approche de l'auditeur : la méthodologie

1. Extraction des données

La réponse

Par l'approche de l'auditeur : la méthodologie

1. Extraction des données

2. Analyse *a posteriori* de ces données

L'existant

État de l'art

- Outils très spécifiques

L'existant

État de l'art

- Outils très spécifiques
- Outils à installer sur les serveurs pour des audits réguliers (à destination des administrateurs souhaitant faire le point régulièrement)

L'existant

État de l'art

- Outils très spécifiques
- Outils à installer sur les serveurs pour des audits réguliers (à destination des administrateurs souhaitant faire le point régulièrement)
- Exclusivement à base de checklist

La solution

Idée

1. Les composants d'une architecture peuvent être modélisés (serveurs, terminaux, pare-feu, routeurs, switchs)

La solution

Idée

1. Les composants d'une architecture peuvent être modélisés (serveurs, terminaux, pare-feu, routeurs, switchs)

2. Il faut s'adapter à la méthodologie en deux phases : extraction puis analyse

L'outil : l'extraction

```
echo "[+] Exécution de la version 0.1 du script d'extraction"
echo $VERSION > $OUTDIR/version_script.txt

echo "[+] liste des fichiers et des droits associés"
### Liste des fichiers avec exclusion de répertoire (Linux)
#find / -type d \( -wholename "/directoryA" -o -wholename "/DirectoryB" \) -prune -o -ls > $OUTDIR/find.txt
find / -ls > $OUTDIR/find.txt

echo "[+] liste des processus en écoute"
netstat -a -n -p > $OUTDIR/netstat.txt

echo "[+] liste des sockets en écoute"
ss -ltp > $OUTDIR/ss-listen.txt

echo "[+] liste de connexions établies"
ss -ptn > $OUTDIR/ss-established.txt

echo "[+] liste des processus actifs"
ps faux > $OUTDIR/ps.txt

echo "[+] liste formatées des processus"
ps -axeo pid,ppid,user,args > $OUTDIR/ps-format.txt

echo "[+] liste des paquets installés"
if [ x$DISTRO = "xdebian" ]; then
    dpkg -l > $OUTDIR/pkg_list.txt
elif [ x$DISTRO = "xredhat" ]; then
    rpm -qa --last > $OUTDIR/pkg_list.txt
else
    echo "[!] attention, distribution non reconnue"
fi

echo "[+] contenu du répertoire /etc/"
### Ignore certains fichiers sensibles (linux)
#tar cjf $OUTDIR/etc.tar.bz2 -p --atime-preserve --wildcards --exclude "/etc/passwd*" --exclude "/etc/shadow*" --exclude "/etc/group*" --exclude "/etc
/krb5.conf" --exclude "/etc/sudoers*" --exclude "/etc/publickeys*" --exclude "/etc/pki*" /etc
```

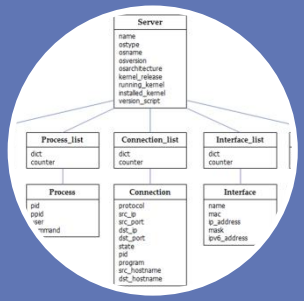
L'outil : le framework

Comment proposer une structure pour manipuler les données facilement et fournir à l'auditeur les données pertinentes

1. Le *framework*

2. Deux manières de l'utiliser
    1. Utilisation en ligne de commande pour l'accès aux données
    2. Le développement de scripts d'analyse

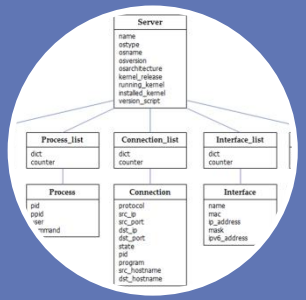
Architecture de l'outil



Modèle de données

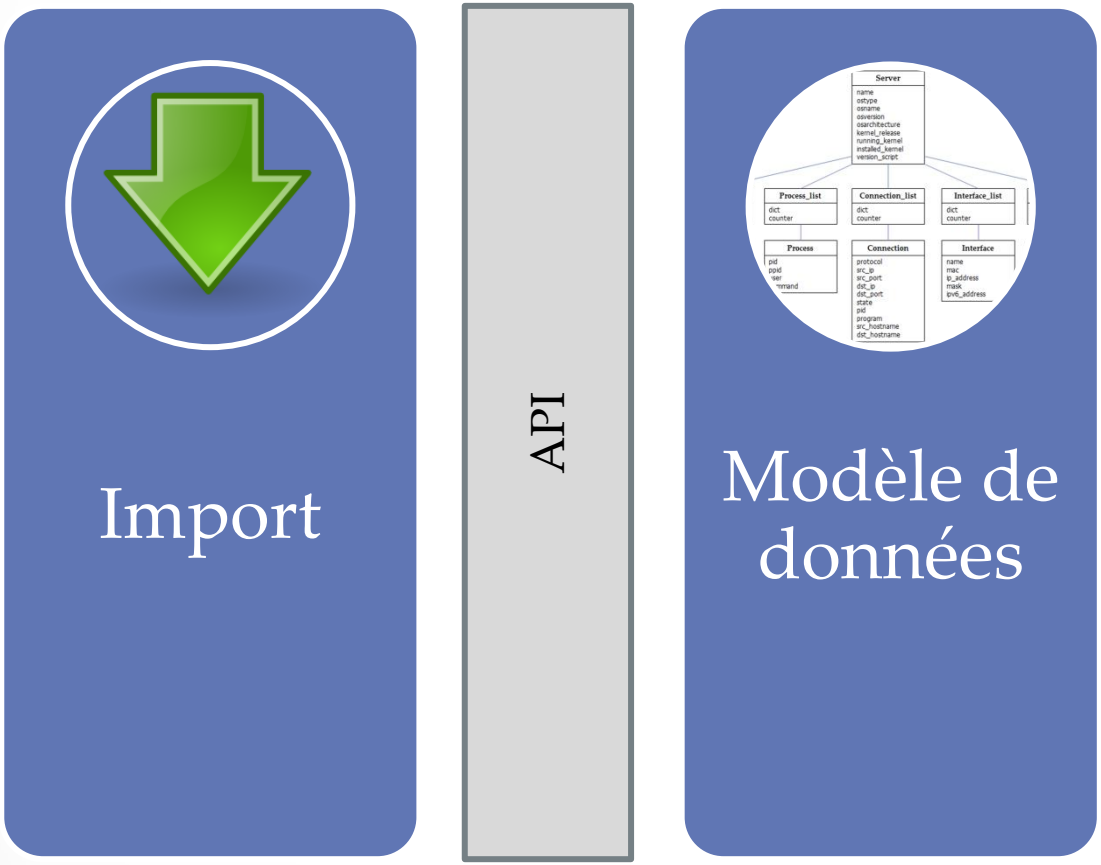
Architecture de l'outil

Import

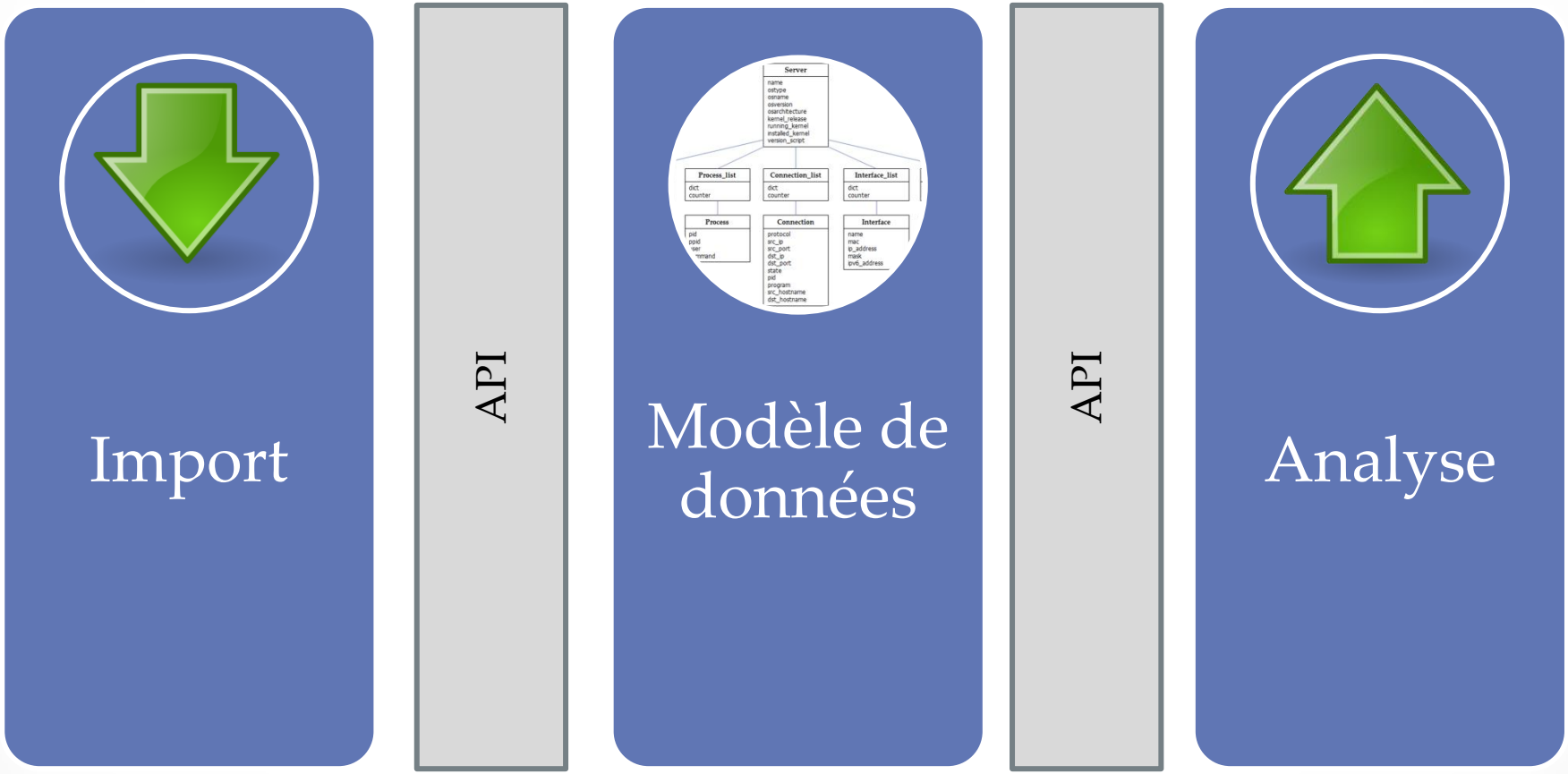


Modèle de données

Architecture de l'outil



Architecture de l'outil



Import d'un serveur

```
In [2]: s = importer.Import_server_from_archive("extract_sample/Debian/wheezy.config.3952.tar.bz2", config)

Server hostname : wheezy
OS type : Debian
Extraction script version : 0.1
OS detected : wheezy
OS version : 7.5
OS architecture : x86_64
Kernel release : 3.2.0-4-amd64
Kernel version (running) : 3.2.57-3+deb7u1
35 accounts imported.
2 interfaces imported.
44 connections imported.
189 processes imported.
259870 files imported.
10 nsswitch rules imported.
10 SSH rules imported.
6 sudoers rules imported.
4 fstab rules imported.
3 cronatb rules imported.
127.0.0.1 ever in the ip_hostname dictionary for the host : wheezy
4 ip_hostname_local imported.
1561 packages imported.
4 pam rules imported.
Server successfully imported. Time : 2.7 secs
```

Processus exécutés par root

```
def listening_connections(self):
    return self.server.connections.filter_connections(state = "LISTEN, ESTABLISHED")

def root_process(self):
    return self.server.processes.filter_process(user="root", hide_ppid="0, 1, 2")
```

Processus exécutés par root

```
def listening_connections(self):
    return self.server.connections.filter_connections(state = "LISTEN, ESTABLISHED")

def root_process(self):
    return self.server.processes.filter_process(user="root", hide_ppid="0, 1, 2")
```

```
def listening_root_process(self):
    ret = sf.ConnectionList()
    listen_ps = self.listening_connections()

    for connection in listen_ps.dict.values():
        ps = self.server.processes.filter_process(pid = str(connection.pid))
        if ps.counter != 1:
            self._logger.error("PID of process after filtering not unique")
            return False
        elif ps.dict[1].user == "root":
            ret.add_connection(connection)
    return ret
```

Processus exécutés par root

```
================================================================================
==                      Processes executed by root                            ==
================================================================================
Short command is showed. Refer to the help of this function

pid   ppid  user               command (short)

670   451   root               udevd                                          1
671   451   root               udevd                                          2
2462  2436  root               /usr/lib/gdm3/gdm-simple-slave                 3
2491  2462  root               /usr/bin/Xorg                                  4
2492  2404  root               /sbin/dhclient                                 5
3507  2462  root               gdm-session-worker                             6
3648  3647  root               udisks-daemon:                                 7
3951  3895  root               sudo                                           8
3952  3951  root               /bin/bash                                      9
3971  3952  root               ps                                             10


================================================================================
==                  Listening processes executed by root                      ==
================================================================================

protocol  src_ip           src_port  dst_ip           dst_port  state         pid       program            src_hostname   dst_hostname
```

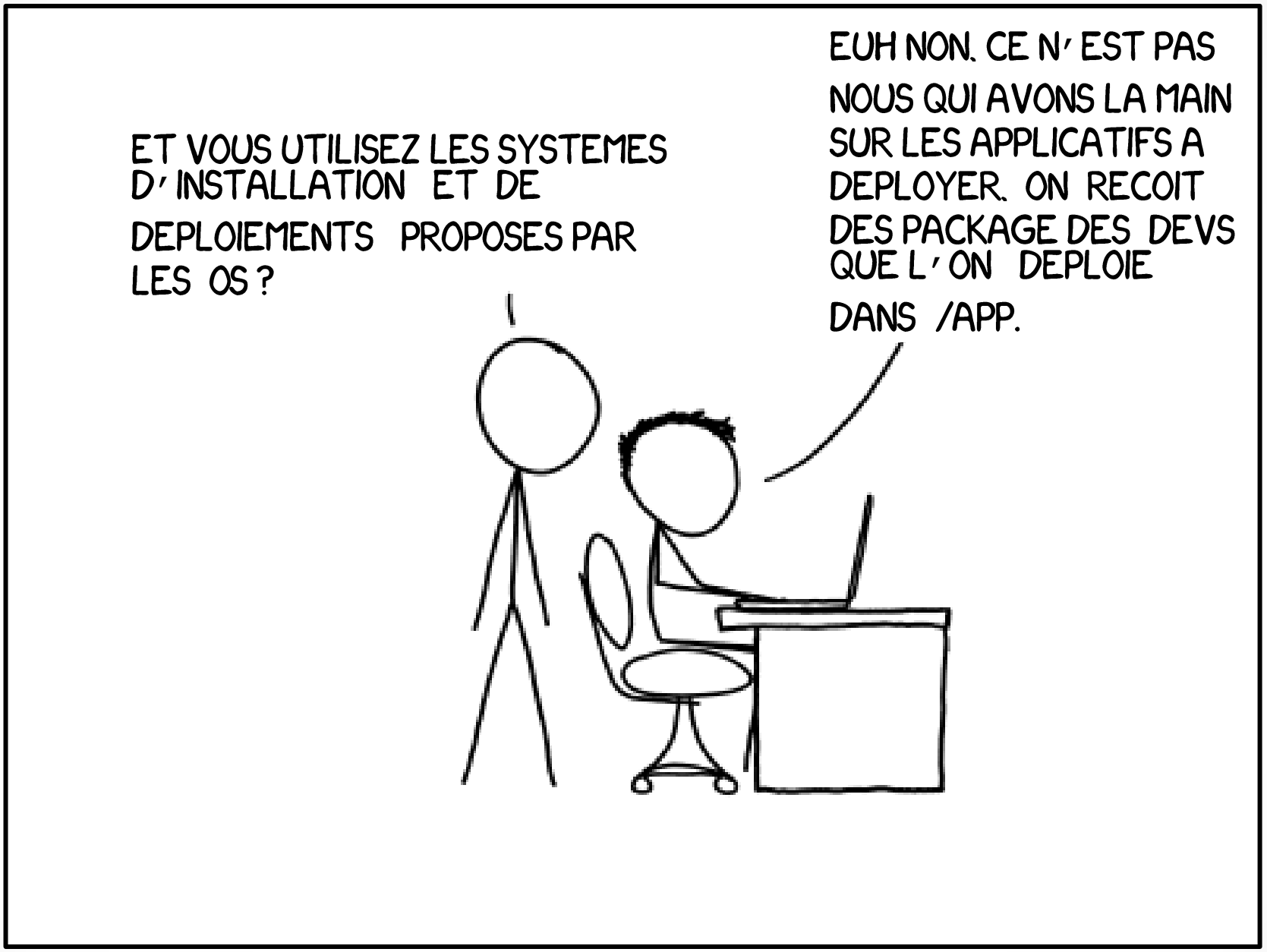
Analyse des versions

```
================================================================================
==                      Kernel analysis results                               ==
================================================================================
Release : 3.2.0-4-amd64 / version running : 3.2.57-3+deb7u1 / version installed : 3.2.57-3+deb7u1 / version up to date : 3.2.68-1+deb7u1
The kernel release is up to date but the version is obsolete. Proceed updates

================================================================================
==                      Packages analysis results                             ==
================================================================================
Packages number on the server : 1561
Packages up to date : [86.93%]
Packages obsolete : [12.94%]
Packages not found on the internet : [0.13%]

================================================================================
==                      Packages not found                                    ==
================================================================================
jenkins                                                 1.563                                   1
libmozjs10d                                             10.0.12esr-1                            2


================================================================================
==                      Packages obsolete                                     ==
================================================================================
acpi-support-base               0.140-5             0.140-5+deb7u3      wheezy          1
apache2.2-bin                   2.2.22-13+deb7u1    2.2.22-13+deb7u4    wheezy          2
apt                             0.9.7.9+deb7u1      0.9.7.9+deb7u7      wheezy          3
apt-utils                       0.9.7.9+deb7u1      0.9.7.9+deb7u7      wheezy          4
at                              3.1.13-2            3.1.13-2+deb7u1     wheezy          5
base-files                      7.1wheezy5          7.1wheezy8          wheezy          6
bash                            4.2+dfsg-0.1        4.2+dfsg-0.1+deb7u3 wheezy          7
bind9-host                      1:9.8.4.dfsg.P1-6+nmu2+deb7u11:9.8.4.dfsg.P1-6+nmu2+deb7u4wheezy            8
binutils                        2.22-8              2.22-8+deb7u2       wheezy          9
bsd-mailx                       8.1.2-0.20111106cvs-18.1.2-0.20111106cvs-1+deb7u1wheezy             10
ca-certificates                 20130119            20130119+deb7u1     wheezy          11
cpio                            2.11+dfsg-0.1       2.11+dfsg-0.1+deb7u1wheezy          12
```

Analyse de configuration OpenSSH

```
================================================================================
==                        SSH analysis results                                ==
================================================================================
3 warnings in SSH configuration
Port                            22              [OK]            [IMPORTED]
Protocol                        2               [OK]            [IMPORTED]
Use privilege separation        yes             [OK]            [IMPORTED]
Log level                       INFO            [OK]            [IMPORTED]
Permit root login               no              [OK]            [IMPORTED]
RSA authentication              yes             [OK]            [IMPORTED]
Pubkey authentication           yes             [OK]            [IMPORTED]
Permit empty password           no              [OK]            [IMPORTED]
Password authentication         yes             [WARNING]       [DEFAULT]
X11 forwarding                  yes             [WARNING]       [IMPORTED]
Use PAM                         yes             [WARNING]       [IMPORTED]
```

ET VOUS UTILISEZ LES SYSTEMES D'INSTALLATION ET DE DEPLOIEMENTS PROPOSES PAR LES OS ?


ET VOUS UTILISEZ LES SYSTEMES D'INSTALLATION ET DE DEPLOIEMENTS PROPOSES PAR LES OS ?
EUH NON. CE N'EST PAS NOUS QUI AVONS LA MAIN SUR LES APPLICATIFS A DEPLOYER. ON RECOIT DES PACKAGE DES DEVS QUE L'ON DEPLOIE DANS /APP.

Évolutions

- Documenter les API
- Compléter les équipements gérés (notamment au niveau réseau)
- SCAP ?
- Apporter une couche visualisation

Merci

Merci à Thierry QUINIOU pour sa participation active au développement

Le code est disponible sur GitHub :
https://github.com/maximeolivier/pyCAF.git

Questions ?