redhat

Mauvaises habitudes

PHP Tour Luxembourg 2015

Présenté par :

Remi Collet

Senior Software Engineer, Red Hat Inc.

PHP developer.

fedora

Sommaire

1. Présentation
2. Mauvaises pratiques - sysadmin
3. Mauvaises pratiques - développeur
4. Questions

fedora

Présentation

Remi Collet

- 1998 : PHP 3.0 user
- 2005 : Remi's RPM repository / LAMP
- 2006 : Fedora contributor (PHP stack)
- 2007 : Fedora PHP co-maintainer
- 2011 : PECL developer
- 2012 : Fedora / Red Hat PHP maintainer
- 2012 : PHP developer

=> http://fr.linkedin.com/in/remicollet

fedora

Remi's RPM Repository

- Cible : Fedora, RHEL, CentOS
- PHP 5.4, 5.5, 5.6, 7.0
- Paquets de base
  - Remplacement (php-*), 1 dépôt par version
- *Software Collections*
  - Installation en parallèle (php##-*)
- ~130 extensions
- Upstream de Fedora / RHEL / RHSCL

  => http://rpms.famillecollet.com/

fedora

Mauvaises pratiques (sysadmin)

Versions

Versions actives

- PHP 5.4 sécurité uniquement (5.4.40)
- PHP 5.5 stable, maintenue (5.5.24)
- PHP 5.6 stable, maintenue (5.6.8)
- Master – développement (7.0.0-dev)

fedora

Versions utilisées

- PHP 4 : 4 % !!
- PHP 5.2 : 23 % !
- PHP 5.3 : 44 % !
- PHP 5.4 : 23 %
- PHP 5.5 : 4 %
- Versions vulnérables : ~80 %

http://blog.ircmaxell.com/2014/12/php-install-statistics.html

http://blog.pascal-martin.fr/post/php-versions-stats-2014-10-en.html

fedora

Sécurité

- Utiliser la dernière version mineure d'une branche maintenue
  - Fedora 20 = PHP 5.5.24
  - Fedora 21/22 = PHP 5.6.8
  - Dépôt tiers : remi, dotdeb..
- Nécessité de tester (BC)
  - Utiliser les RC
  - Koschei (CI)

Sécurité

- Utiliser les paquets maintenus d'une distribution « entreprise »
  - RHEL-6.6 : 5.3.3
  - RHEL-7.1 : 5.4.16
  - RHSCL-2.0 : 5.4.40, 5.5.21, 5.6.5
  - Debian Wheezy : 5.4.39
  - Debian Jessie : 5.6.7

=> http://www.redhat.com/advice/speaks_backport.html

fedora

MySQL

Pilote MySQL

- Pilote MySQL Oracle (libmysqlclient)
  - Licence incompatible (GPL, FOSS exception)
  - Code non maintenu
  - Mémoire non limitée (x2)
  - Dépendance à MySQL
- Extensions
  - mysql (PHP < 7), mysqli, pdo_mysql
- Paquets
  - **php-mysql** (rpm) ou php5-mysql (deb)
  - Supprimé de Fedora

fedora

Pilote MySQL

- Pilote MySQL Natif (MySQLnd)
  - Licence PHP
  - Recommandé par le projet
  - Mémoire gérée par PHP
  - Pas de connexion « old_password »
- Extensions
  - mysql (PHP < 7), mysqli, pdo_mysql
- Paquets (fournissent les même extensions)
  - **php-mysqlnd** (rpm) ou php5-mysqlnd (deb)

mod_php

mod_php

- mod_php
  - « Apache HTTPD Server » uniquement
  - Mode *worker* uniquement (processus)
    - ZTS existe mais fortement déconseillé
  - Processus commun (sécurité)
    - CVE-2014-4721 info leak in phpinfo
  - 1 seule version de PHP
    - php5_module / php7_module impossible

```
SetHandler application/x-httpd-php
```

fedora

mod_php / FPM

- php-fpm
  - Apache (2.2/2.4), Nginx, Lighttp...
  - Frontal web multi-thread
  - Isolation
  - version par hôte / projet
  - Docker

```
ProxyPass ...

SetHandler proxy:fcgi://127.0.0.1:9000
SetHandler proxy:fcgi://php-fpm
SetHandler
  proxy:unix:/run/php-fpm/www.sock|fcgi://localhost
```

mod_php / FPM

- Développement, 1 vhost par version

```
<VirtualHost *:80>
    ServerName php56
    <FilesMatch \.php$>
        SetHandler "proxy:fcgi://127.0.0.1:9056"
    </FilesMatch>
</VirtualHost>

<VirtualHost *:80>
    ServerName php70
    <FilesMatch \.php$>
        SetHandler "proxy:fcgi://127.0.0.1:9070"
    </FilesMatch>
</VirtualHost>
```

mod_php / FPM

- Dockerfile

```
FROM centos:6

RUN yum -y update && yum clean all

RUN yum -y install php-fpm php-mbstring php-mysql php-gd &&
yum clean all

RUN sed -e 's/127.0.0.1:9000/9000/' \
        -e '/allowed_clients/d' \
        -e '/catch_workers_output/s/^;//' \
        -e '/error_log/d' \
        -i /etc/php-fpm.d/www.conf

RUN mkdir -p /var/www/html

ENTRYPOINT /usr/sbin/php-fpm --nodaemonize
```

Configuration

php.ini

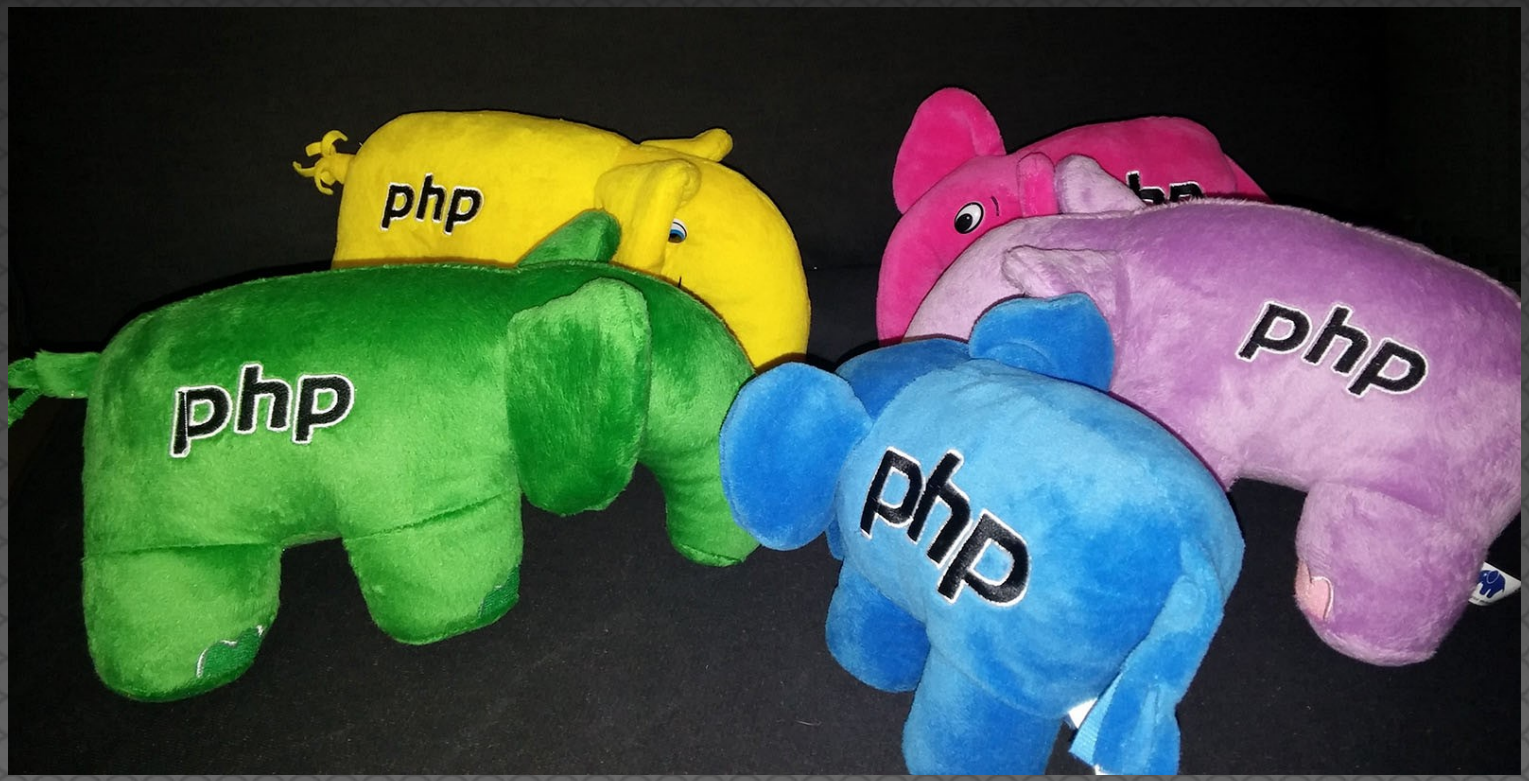
- php.ini
  - Global pour le système
  - Fournit par la distribution
- Global => /etc/php.d/*.ini
- mod_php
  - php_value dans httpd.conf / .htaccess
- php-fpm
  - php_value dans <pool>.conf
- .user.ini

En vrac

- Apache propriétaire des scripts PHP
  - httpd-itk
- AddHandler php5-script .php
- Désactiver SELinux
- yum install php*
- APC
- On ne change pas quelque chose qui marche
- Composer (duplication, mise à jour, ...)
- ...

fedora

Mauvaise pratiques (développeurs)



(un)serialize

unserialize

- Conçu pour stockage **local** et **temporaire** de données générées par « serialize »
  - Ex : session
- Données dépendantes de la version PHP
- Jamais de stockage longue durée (DB)
- Jamais d'échange inter-applications
- Jamais de données fournies par l'utilisateur
- Problème de sécurité (__wakeup)
- Utiliser un autre serializer (json)

unserialize - CVE

- Vulnerabilités récentes (<6 mois)
  - CVE-2014-3669 integer overflow
  - CVE-2014-8142 use after free
  - CVE-2015-0231 use after free
  - CVE-2015-0273 use after free (date)
  - #69085 type confusion (soap)
  - CVE-2015-2787 use after free
  - d'autres prochainement...
- Utiliser les données sérialisées sur le réseau, c'est rendre possible l'exploitation de ces CVE.

unserialize - __wakeup

```
class MiniCache implements Serializable {
    private $path, $file;
    function __construct($path) {
        $this->file =
          file_get_contents($this->path = $path);       }
    function get() {
        return $this->file;                             }
    function serialize() {
        return serialize($this->path);                  }
    function unserialize($data) {
      $this->file =
        file_get_contents($this->path = unserialize($data));
    }
}
```

unserialize - __wakeup

- Comment vérifier les entrées ?
  - "C:9:"MiniCache":19:{s:11:"/etc/passwd";}"
- Pas de contrôle sur les classes (php < 7)
- Pas de contrôle sur les propriétés
- Execution de code via __wakeup
- PHP 7: Filtered unserialize()
  - https://wiki.php.net/rfc/secure_unserialize

Extensions

POSIX regex

- Extension « *ereg* » dépréciée (5.3)
- Supprimée en 7.0
- Mauvaises performances
- Fonctions
  - Ereg_*
  - Split*
- => utiliser les expression rationnelles compatibles perl (extension « *pcre* », fonctions preg_*)

API MySQL originale

- Extension « *mysql* »
  - dépréciée en 5.5
  - supprimée en 7.0 (pecl ?)
- Maintenue a minima
- Uniquement procédurale
- Fonctions mysql_*
- Fontion mysql_escape_string : Danger !
  - Non liée à la connexion (charset)

API MySQL

- Utiliser MySQL Improved (« *mysqli* »)
  - Fonctions et classes mysqli_*
  - Méthode mysqli::real_escape_string
    - Liée à la connexion
  - Portage trivial
- Utiliser PHP Data Objects (« *PDO* »)
  - Abstraction au moteur
  - Portage plus couteux

IMAP, POP3 et NNTP

- Extension « imap »
- Nécessite la bibliothèque *libc-client* (*uw-imap)* non maintenue
- Conservée en PHP 7.0 :(
- Utiliser une bibliothèque pure-PHP
  - roundcubemail
  - horde
  - *...*

fedora

Mcrypt

- Extension « mcrypt »
- Nécessite la bibliothèque *libmcrypt* non maintenue depuis >7 ans
- Conservée en PHP 7.0 :(
- Utiliser la fonction « crypt »
- Utiliser l'extension « openssl »
- Voir « Fonctions de hachage de mot de passe »
  - PHP 5.5+
  - PHP 5.3+ https://github.com/ircmaxell/password_compat

Mcrypt - example

- Roundcubemail (TripleDES)

```
if (function_exists('openssl_encrypt')) {
  …
  $cipher = $iv . openssl_encrypt(
                $clear, $method, $ckey, $opts, $iv);
    …
} else if (function_exists('mcrypt_module_open') &&
  ($td = mcrypt_module_open(
       MCRYPT_TripleDES, "", MCRYPT_MODE_CBC, ""))) {
   …
} else {
  /* pure PHP implementation */
}
```

fedora

Mcrypt - example

- Phpseclib (Random)

```
if (function_exists('openssl_random_pseudo_bytes')) {
  return openssl_random_pseudo_bytes($length);
}
$fp = @fopen('/dev/urandom', 'rb');
if ($fp) {
  return fread($fp, $length);
}
if (function_exists('mcrypt_create_iv')) {
  return mcrypt_create_iv($length, MCRYPT_DEV_URANDOM);
}
/* pure PHP implementation */
...
```

Password - example

- password_hash / password_verify

```
function set($pass) {
  $this->save(password_hash($pass, PASSWORD_DEFAULT));
}
function check($input) {
  $real = $this->load();
  if (password_verify($input, $real)) {
    if (password_needs_rehash($real, PASSWORD_DEFAULT)) {
      $this->set($input);
    }
    return true;
  }
  return false;
}
```

Autres extensions

- mssql
- sybase_ct

Commentaires

https://joind.in/14283

fedora

Questions?

Contact:
remi@php.net