

Bonnes pratiques



en développement web

Vincent Mazenod

⚠ Ingénieur d'Etudes au CNRS

💻 Développeur

⚙️ En poste au [LIMOS](#)

🏢 Bureau A205 - 2^{ème} étage

📞 04 73 40 50 41

✉️ vincent.mazenod@isima.fr

🏆 Expert [SSI](#) à la [CRSSI DR7 CNRS](#)

💼 [cv](#)

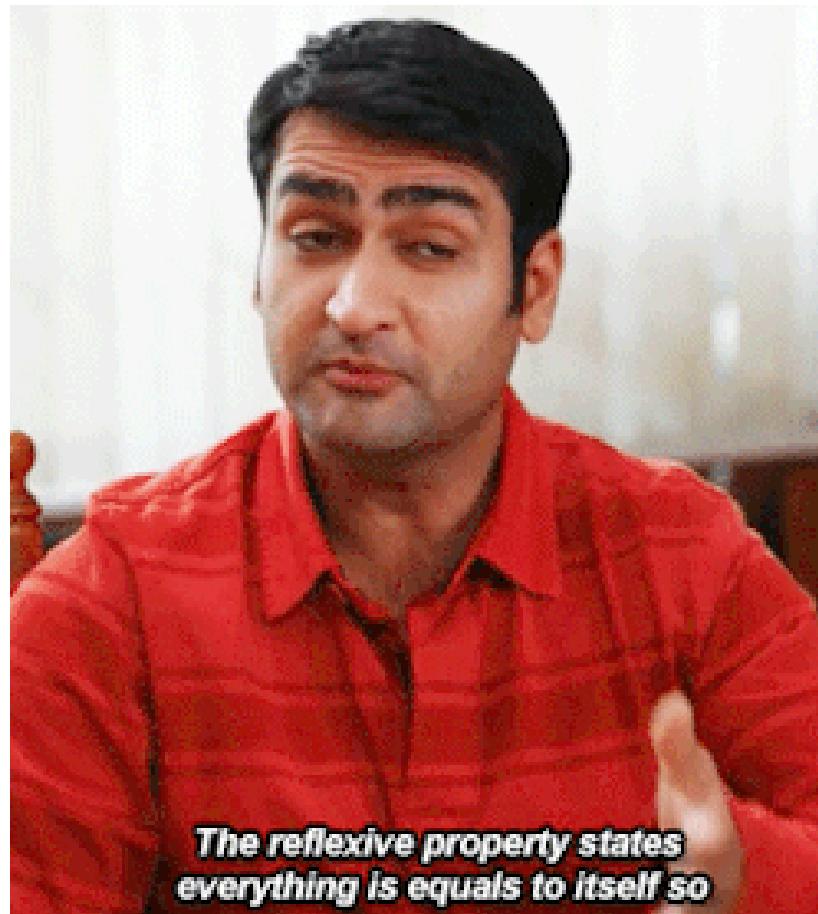
📢 [blog](#)

⚙️ [slides](#)

🐦 📱 🌐 📺



Dev



***The reflexive property states
everything is equals to itself so***

- client des dernières technologies
- casse tout et recommence
- teste son propre code
- le vendredi soir
 - ship la dernière feature et par en week-end

Ops



- systèmes hétérogènes
- continuité de services
- le vendredi soir
 - prie pour que le dernier déploiement ne crashe pas

Problèmes communs



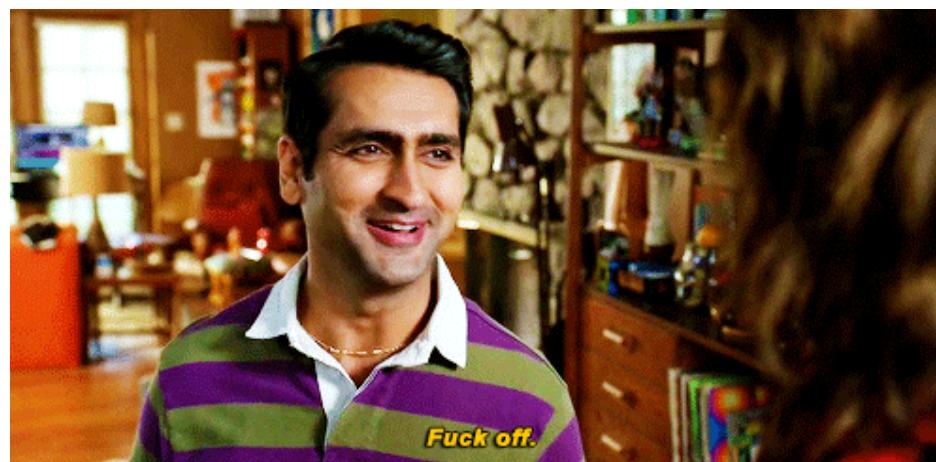
pour vision divergentes

- fonctionnalité **VS** sécurité
- nouveauté **VS** stabilité
- création **VS** maintenance
- hétérogénéité **VS** environnement
- tests antirégressions **VS** mise en production

Conflits



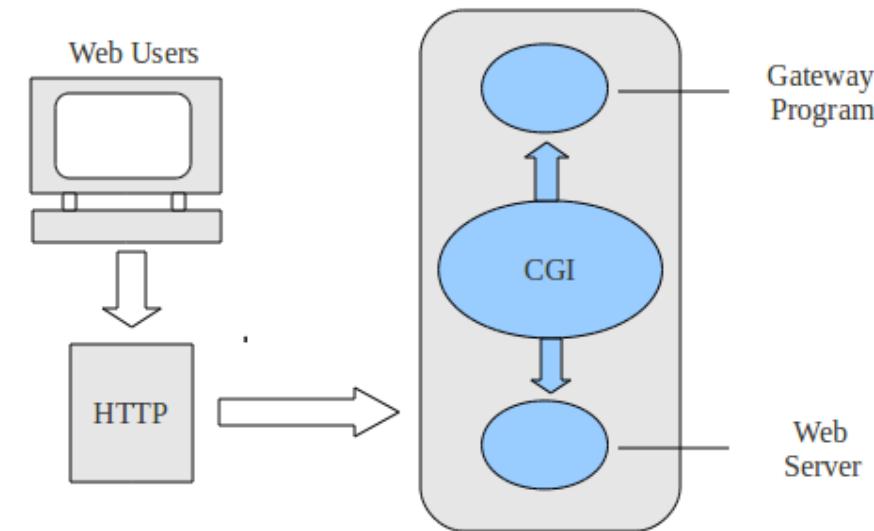
Do I be honest or nice?



Me Myself and I

```
// life motto
if (sad() === true) {
  sad().stop();
  beAwesome();
}
```

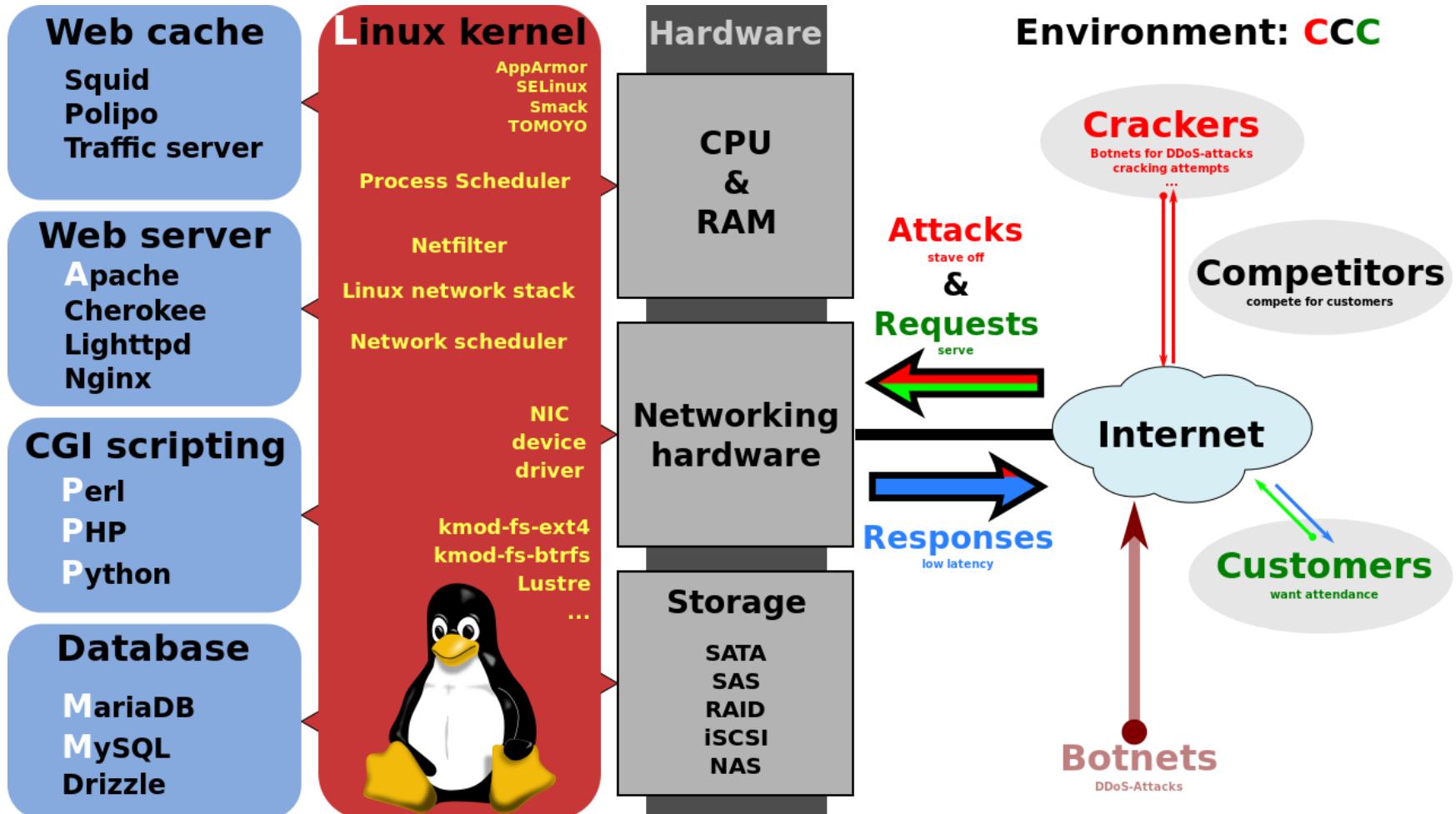
2001 / CGI / Perl / cpan



<http://oldschool.com/helloWorld.cgi>

```
#!/usr/local/bin/perl
print "Content-type: text/html\n\n";
print "Hello World.\n";
print "Heres the form info:<P>\n";
my($buffer);
my(@pairs);
my($pair);
read(STDIN,$buffer,$ENV{'CONTENT_LENGTH'});
@pairs = split(/&/, $buffer);
foreach $pair (@pairs)
{
  print "$pair<BR>\n"
}
```

2001 / LAMP



ispconfig?

2001 / Best practice

The screenshot shows the UltraEdit-32 interface with the following details:

- Title Bar:** UltraEdit-32 - myfiles - [C:\myfiles\cftopen.cpp]
- Menu Bar:** File, Edit, Search, Project, View, Format, Column, Macro, Advanced, Window, Help
- Toolbar:** Standard file operations like Open, Save, Find, Copy, Paste, etc.
- Tab Bar:** randomizer.js, abs_space.java, cftopen.cpp (active), config.cgi, datasec.c, formmail.pl, index.html
- Left Panel:** Explorer view showing a tree structure of drives (A:, C:, D:, U:, V:, W:, X:, Y:, Z:) and an FTP Accounts section with a Dev folder containing subfolders like /public_html/ and files such as index.html, mail_handler., readme.txt, rssfeed.xml, and header.inc.pl.
- Code Editor:** The main window displays C++ code for a SortFunc callback function. The code includes declarations for threadStruct, CFTPOpen*, CListCtrl*, LV_FINDINFO, and LV_ITEM, along with several if statements and variable definitions.
- Right Panel:** A list of functions or symbols related to the current file, including CFTPOpen, DoDataExchange, BEGIN_MESSAGE_MAP, ON_BN_CLICKED, END_MESSAGE_MAP, SetDialogBox, SortFunc (which is highlighted in blue), SFTPDone, SFTPFileStatus, SFTPAtributes, FTPOperationProc, ParseString, GetParseStringPieces, CvtBinaryToAsciiHex, CvtBAsciiHexToBinary, ect, dcp, DirExists, CreateMyDir, WriteDirList, LoadAccountList, and LoadAccount.
- Bottom Panel:** A search results window titled "Find 'SortFunc' in 'C:\myfiles\cftopen.cpp'" showing three occurrences of the SortFunc declaration and definition within the file.
- Status Bar:** For Help, press F1, Ln 244, Col. 29, C0, DOS, Mod: 4/1/2005 3:22:48PM, Bytes Sel: 8, INS.

2001 / WYSIWYG

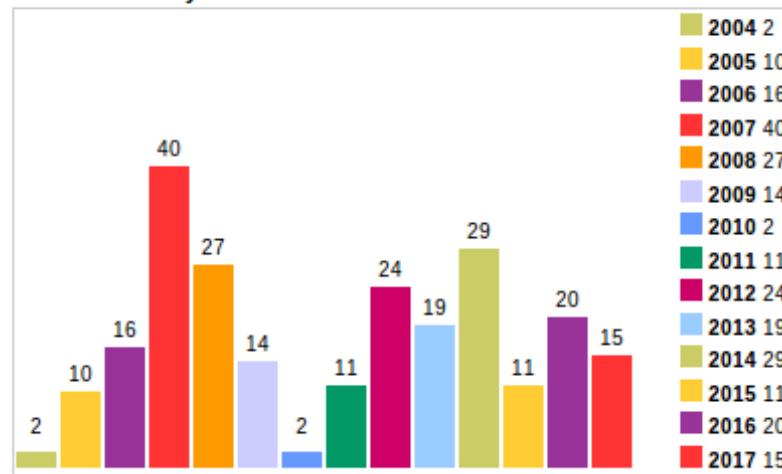


- compatibilité Netscape 4.7
- compatibilité IE 5/6

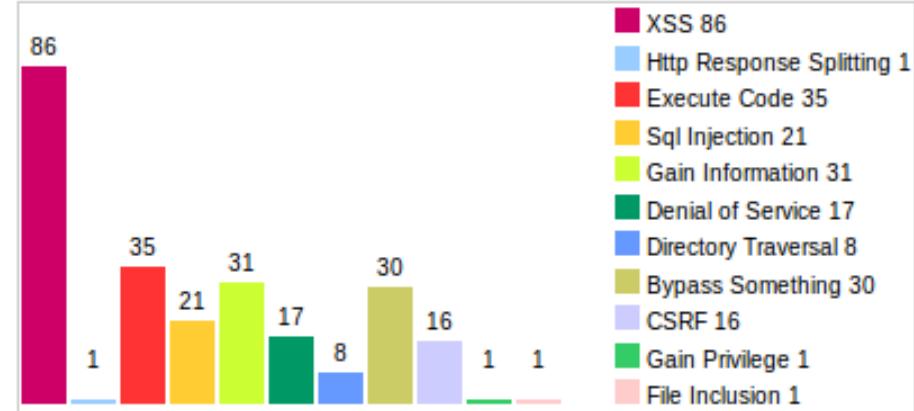
2007 / CSS frameworks?

2001 / CMS

Vulnerabilities By Year



Vulnerabilities By Type



stats pour wordpress

- 2001 / [spip](#) par [uZine](#)
- 2002 / [drupal](#)
- 2004 / [wordpress](#)
- 2005 / [joomla](#) ...

2003 / Ajax & JavaScript



- XMLHttpRequest
- ActiveXObject("Msxml2.XMLHTTP")
 - [sarissa](#)
 - 2005 / Prototype / [script.aculo.us](#)
 - 2006 / MooTools
 - 2006 / JQuery

2004 / Frameworks



- 2004 / Ruby On Rails
- 2006 / symfony (1.0)
- 2011 / Laravel
- 2012 / composer
- 2017 / symfony (3.3)

Frameworks

- structure de fichiers cohérente
 - [A7-Missing Function Level Access Control](#)
- configuration
 - [A5-Security Misconfiguration](#)
 - firewall [A4-Insecure Direct Object Reference](#)
- mvc
 - moteur de template
 - [A3-Cross-Site Scripting \(XSS\)](#)
 - sous framework de formulaire
 - [A8-Cross-Site Request Forgery \(CSRF\)](#)
- environnements
 - préserve la prod
 - [A6-Sensitive Data Exposure](#)
- orm
 - requêtes préparées
 - [A1-Injection](#)

Frameworks

- plugins
 - utilisateurs
 - [A2-Broken Authentication and Session Management](#)
 - commandes systèmes & filesystem
 - [A1-Injection](#)
 - d'upload
 - [A6-Sensitive Data Exposure](#)
- cli
 - [A9-Using Components with Known Vulnerabilities](#)
- tests
 - fonctionnels
 - unitaires
 - fixtures
- déploiement

100% [OWASP Top 10](#)

le cli

- composer (php)
- RubyGems (Ruby)
- pip (Python)
- npm (nodejs)



How one developer just broke Node, Babel and thousands of projects in 11 lines of JavaScript

2008 / github

- remplace sourceforge (~1999) en quelques mois
 - ne distribue pas des versions packagées
 - distribue le code et rien que le code
 - permet d'explorer le code en ligne
 - invente les **pull requests**
 - branch & tag
 - bug tracker
 - wiki doc
 - git*pages
 - statistiques de popularité, de vie
 - timeline des dernières modifs
 - Markdown à tous les étages

l'opensource a gagné

le libre pas forcément

2011 / gitlab

-  gitlab =  github + 
 -  gitlab.isima.fr = gitlab/gitlab-ce on CoreOS on OpenStack
- repo
 - team
 - private / protected /public
- python-gitlab <3
- 2013 / gitlab ci
 - runner docker
 - .gitlab.yml

extrait gitlab-ci

CI / CD

- variable
 - ssh key
 - username
 - passwords
- multi stage
 - pre-script
 - build
 - deploy
- pastille verte / rouge
 - s'arrête au moindre problème

maquetter son déploiement

- vagrant
 -  [gitlab](#)
- ansible
 -  [gitlab](#)

Dev<3Ops



Twelve-Factor App!