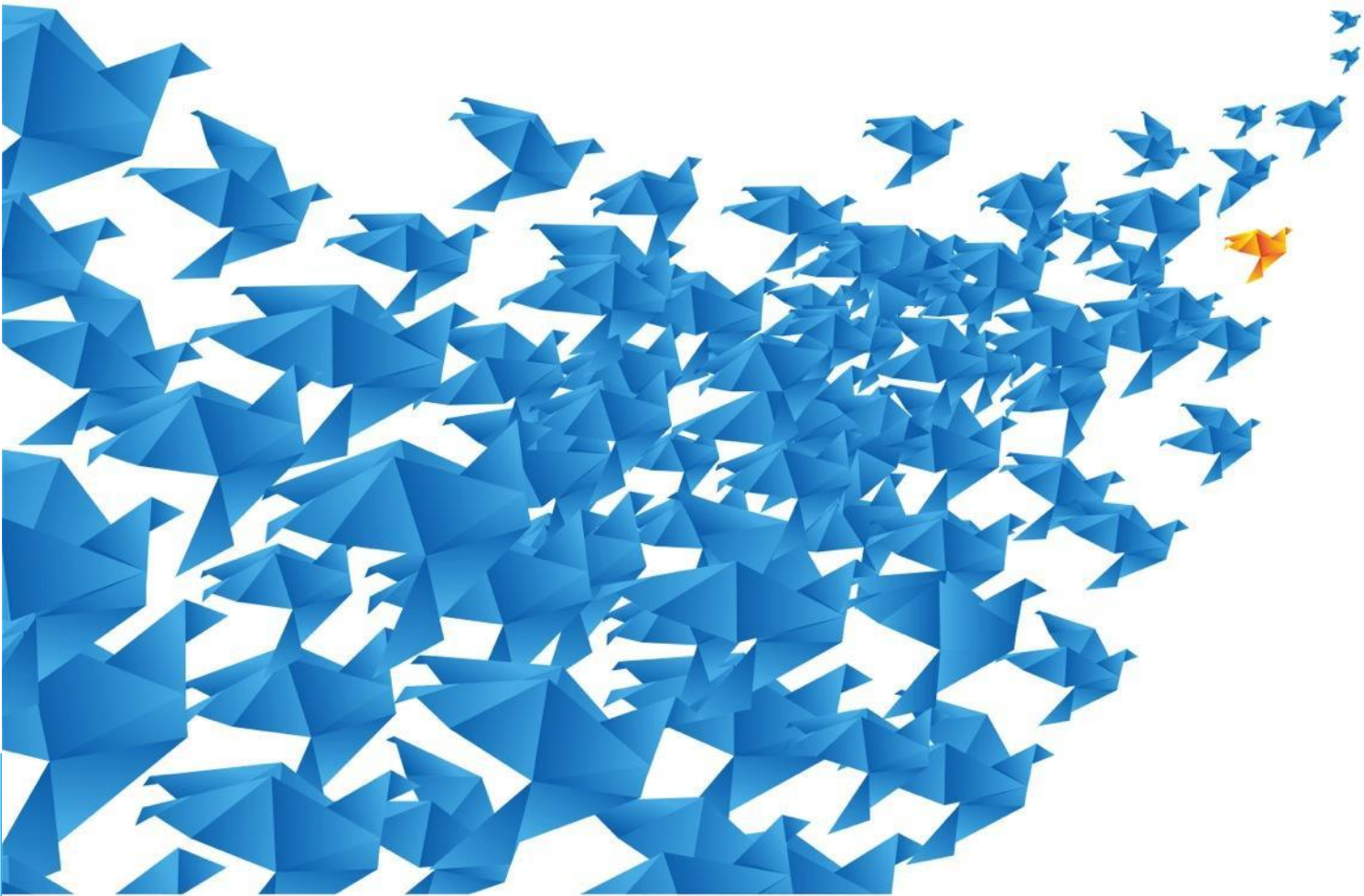
Acquia®

DREAM IT. DRUPAL IT.



Sécuriser un site Drupal

Meetup Drupal Lyon - 03/07/2014

@AurelienNavarre / @DrupalFacile



Acquia

En un mot...

N'importe quel site semble sécurisé jusqu'au moment où il sera effectivement compromis.

Votre mission est de retarder ce moment et de protéger vos clients du mieux que vous le pouvez.

Cela arrive même aux meilleurs.

Quelques exemples depuis début 2014...















KICKSTARTER

- Mot de passe
- Numéro de téléphone
- Adresse e-mail



- 4.6 de comptes piratés
- Identifiant utilisateur
- Numéro de téléphone



- 200M de comptes piratés
- Identifiant utilisateur
- Etat civil
- Adresse
- Téléphone

Forbes

- Mot de passe
- Numéro de téléphone
- Adresse e-mail



- Identifiant utilisateur
- Mot de passe
- Adresse e-mail





- DDoS massives et successives
- Interruption complète de service
- Demande de rançon !

Quelques types d'attaques...

- **CSRF**: vol de cookie, détournement HTTP
- **XSS**: injection de code dans une page
- **SQL Injection**: requête SQL dans un champ de formulaire non protégé
- **Access bypass**: infos privées accessibles
- **(Distr.) Denial of Service**: faire tomber un site par une attaque coordonnée et ciblée

il y en a bien plus !

Comment arrive une attaque ?

- SA-CORE / SA-CONTRIB non suivis
- Permissions du site trop ouvertes
- Mauvaises pratiques (PHP Filter, hacks...)
- Attaque ciblée (Brute Force, XSS…)
- Identifiants compromis (admin/azerty)
- Serveur(s) vulnérable(s)
- Sauvegardes non-encryptées et volées ou partagées avec des infos confidentielles

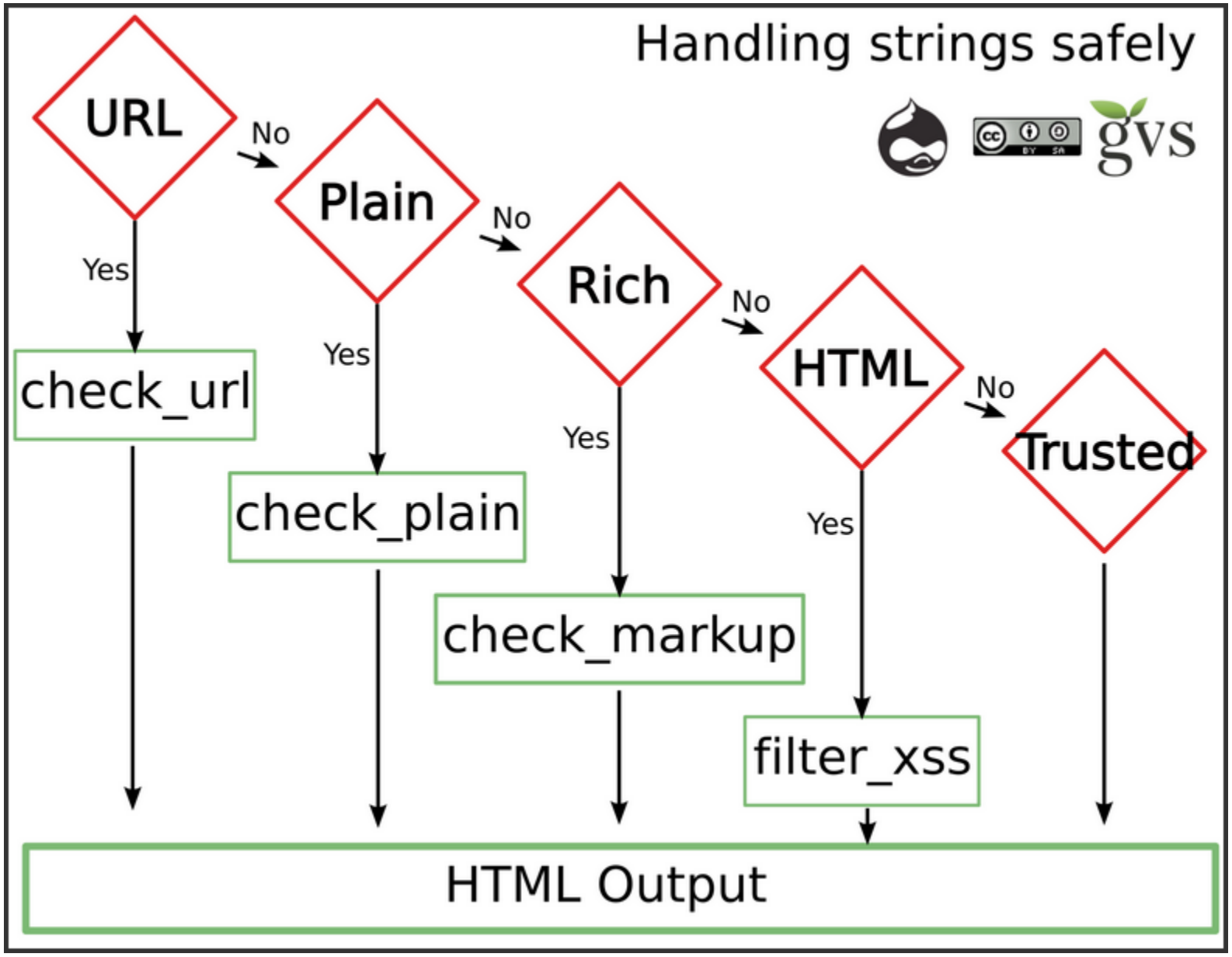
Les plus de Drupal 7

- Meilleur stockage des mots de passe en base de données (SHA512)
- Prévention des attaques XSS et SQL
- Prévention du “flooding”
- Accès restreint à cron.php (clé unique)
- Prévention (D)DoS sur les miniatures d’images (tokens) depuis 7.20

API Drupal

- check_url: nettoie les tags d'une URL
- check_plain*: échap. texte brut vers HTML
- check_markup: texte riche vers HTML
- filter_xss*: filtre le HTML
- Fonctions **preprocess*** (theming)
- Utilisez Form API et Database API (PDO)

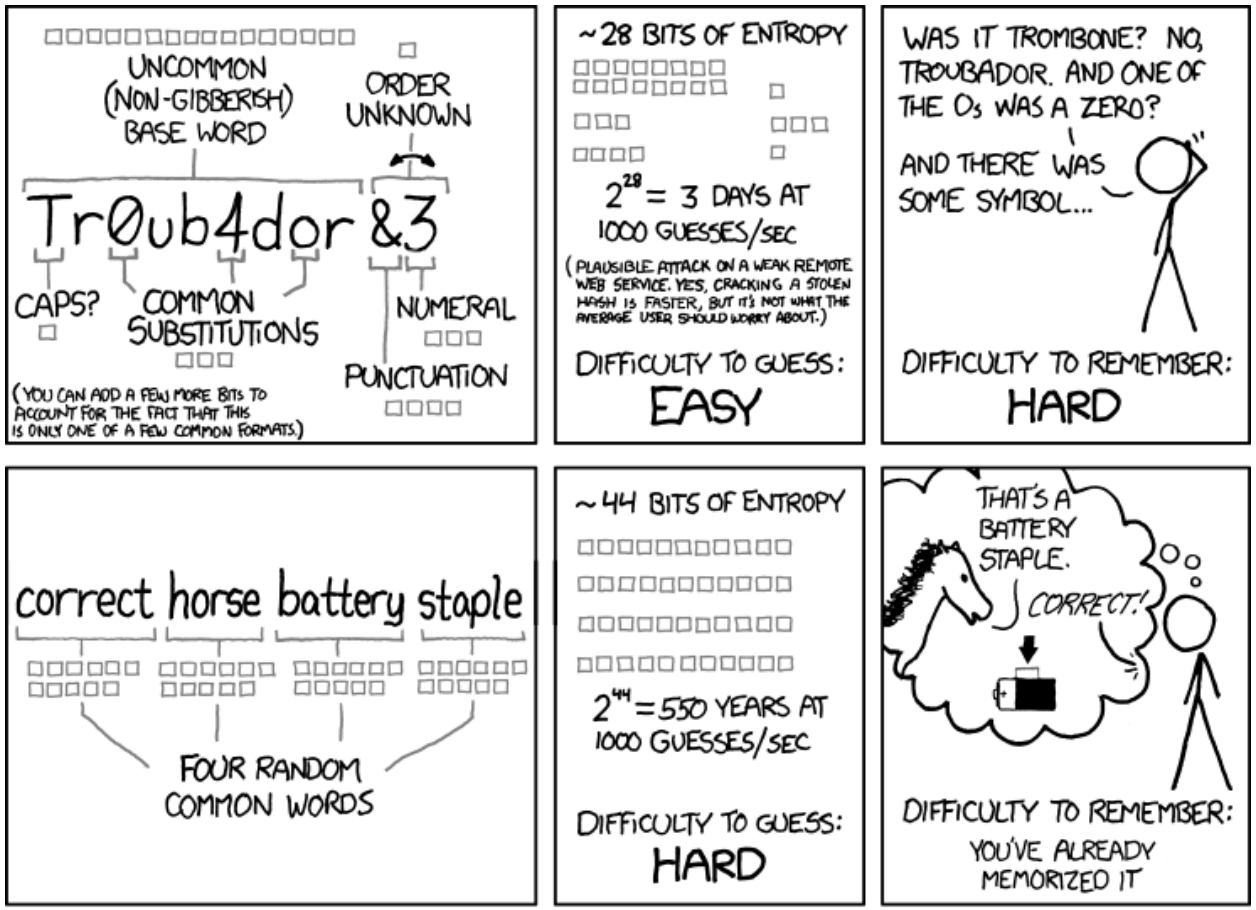
* supprimé et/ou modifié dans Drupal 8


Handling strings safely
URL
No
Plain
No
Rich
No
HTML
No
Trusted
Yes
Yes
Yes
Yes
check_url
check_plain
check_markup
filter_xss
HTML Output

Quelques modules

- Security Review: audit de sécurité
- Security Kit: hardening avancé
- Password Policy: force l'utilisation de mots de passe complexes (pas suffisant !)
- Permissions Lock: permissions renforcées
- Username Enumeration Prevention: évite les attaques par dictionnaire
- Secure Pages: force une connexion SSL

C’est quoi un bon mot de passe ?



http://xkcd.com/936

Acquia

Tester un mot de passe

```
password:                Tr0ub4dour&3
entropy:                 30.435
crack time (seconds):    72600.71
crack time (display):    22 hours
score from 0 to 4:       2
calculation time (ms):   1
match sequence:
'Tr0ub4dour'
pattern:        dictionary
entropy:        17.295
dict-name:      englis
rank:           20111
base-entropy:   14.295
upper-entropy:  1
l33t-entropy:   2
l33t subs:      0 -> o; 4 -> a
un-l33ted:      troubadour
'&3'
pattern:        bruteforce
```

```
password:                correcthorsebatterystaple
entropy:                 45.212
crack time (seconds):    2037200406.475
crack time (display):    65 years
score from 0 to 4:       4
calculation time (ms):   2
```

Fichiers et dossiers

- Fichiers (644)
- /default (755)
- /default/files (744 ou 755)
- /default/themes (755)
- /default/modules (755)
- /default/settings.php (444)

www-data:sitename

Dans un monde idéal

- **HTTPS**, **SSH** et **SFTP** en standard
- **Git** pour le contrôle de version
- Encryptez les données utilisateurs : modules AES, dbee, Encryption...
- **2FA**: de plus en plus utilisé et populaire
- Stratégie **fail-over**: HA, active/hot-spare
- Protections DDoS, CDN, multi-region
- Penetration testing et Load testing

Et Drupal 8 là-dedans ?

- **Module PHP** enfin supprimé du coeur !
- **Twig** : moteur de template sécurisé par défaut via sa fonction d'autoescaping
- **CKEditor** intégré dans le coeur
- **Local image filter** pour limiter l'utilisation du tag <*img*> au nom de domaine courant
- Protection **CSRF** par défaut via le composant Routing de Symfony

Si vous savez...

- Rester informé (RSS, Twitter…)
- Installer et configurer des modules
- Suivre les bonnes pratiques Drupal
- Limiter le code PHP dans votre thème
- Utiliser l'API Drupal et des modules contrib
- Suivre le principe KISS

Alors vous avez fait 98% du travail !

Et les 2% restants alors ?

- Expertise Drupal
- Expertise PHP
- Expertise système et réseau
- Expertise en sécurité
- ...

...beaucoup d'expertise pour 2%

Comment gérer un piratage ?

- Communiquez rapidement via les réseaux sociaux...et soyez transparent !
- Forcez la réinitialisation des mots de passe
- Analysez vos logs système / Drupal
- Réparer la faille est le strict minimum. Allez plus loin et devenez un exemple à suivre !
- Module Paranoia...si vous le devenez.

...apprenez de vos erreurs !

Quelques ressources

- https://www.drupal.org/writing-secure-code
- https://drupal.org/security-team
- https://drupal.org/security/secure-configuration
- https://github.com/greggles/vulnerable
- http://crackingdrupal.com
- http://www.ubuntu.com/usn/

Merci. Questions ?

Acquia®

DREAM IT. **DRUPAL IT.**