Session sécurité

Session sécurité

Damien Tournaud

Enjeu : concept de haut niveau pour comprendre d'où viennent les problèmes de sécurité
Session "haut niveau" - assez éloignée des API drupal

Pas en arrière pour essayer de comprendre la cause des systèmes de sécurité et pourquoi il faut les résoudre

Règles de sécu de base
- installer un petit nb de modules
- installer modules stables et maintenus
- maj les modules
- faire attention droits - en particulier aux formats d'entrée (jamais full html à personne)
- jamais ftp
- modifier mdp régulièrement

Reco drupal sécurité & bonne practice des api drupal : voir drupal.org

Drupal a une équipe de sécurité dédiée depuis 2005 (que des bénévoles) - petit budget de la part de l'asso pour des restos, mais c'est tout - certains employeurs laissent du temps à leurs employé sur leur temps de travail pour participer à la communauté.
- coordonner les releases de sécu de core (3 en 2013), modules contrib (93) -> veille sur les issues de sécurité et répond aux failles signalées par les utilisateurs.
- veille, documentation, formation sur les problèmes de sécurité auprès d'un plus grand public (comme aujourdhui)

PLAN : 3 types de vulnérabilité
1/ contournement d'accès
2/ détournement du navigateur
3/ contournement d'application

pbm de la doc aujourd'hui sur les problématiques de sécurité : rapidement tres (trop) technique. Ici, on essaie de rester assez générique.

1/ contournement d'accès
l'utilisateur est capable de faire des choses qu'il ne devrait pas être capable - vulnérabilité est un contournement d'une règle métier.
règle business spécifique à un site - erreur de developpement oublier de masquer les trucs non publiés), ou de permission

2/ détournement du navigateur
un utilisateur légitime est forcé à faire qqchose parce que son navigateur a été détourné.
a) les pbmtiques d'interception : si connexion non secure (lieu publique), il est facile d'intercepter cookie, nom d'user, mdp etc...
solution : utiliser ssl (https) - FULL HTTPS (y compris pour les utilisateurs anonymes).
attention : secure pages cookie transmissibles en http sont interceptables -> moins de sécurité que du full https. (gestion token pas ok sur drupal 7 core et secure pages).

voir https://drupal.org/project/securepages et les 2 patches à appliquer à l'install du module

marker les cookies comme "https only"

b) les navigateurs executent du code arbitraire. Html est maintenant un langage de code.
cross site scripting (xss) ie injection de code - vulnérabilité la plus commune sur internet (>80%). Très facile à faire (exemple dans les slides de Damien).
2 endroits critiques : à l'intérieur d'une balise et à l'intérieur d'un attribut
-> traitement particulier nécessaire

check plain : transforme du plain text en htlml
 filter xxs et filter xxs admin : filtre un snipper qui ressemble à du html pour supprimer tout ce qui peut etre exécutable
Ces fonctions ne sont pas "primairement" des fonctions de sécurité. C'est une bonne practice quoiqu'il en soit. Il se trouve que par effet de bord, ça supprime les problemes de sécurité.

3/ contournement d'application
SAME ORIGIN policy
2 origines différentes n'ont pas accès à leur dom, et n'ont pas la possibilité de faire de requete http de l'une à l'autre.
Les plugin tierce parties ne sont pas directement soumis à la regle de same origine - typiquement flash est un gros pbm de sécurité parce qu'il n'implemente pas exactement la regle de meme origine.
Cross site request forgery (via get)
la norme http spécifie qu'une requete en get est sensée juste ramener du contenu - pas faire de modification - si on respecte ça, on ne peut pas modifier via un get, tout va bien.
utilisation de token dans les formulaires - pour s'en protéger, utiliser l'API de drupal.

Nouvelle série d'attaque basée sur les plugins navigateurs d'upload.
2 grandes classes de pbm :
 (escalation ?)
 si le serveur renvoie un type de contenu (ex: une image jpg), certains navigateurs essaient de deviner le type du contenu (??). (issue jusqu'à IE8, IE9 pas safe mais mieux. IE10 enfin ok).
 balises objet et vulnérabilité flash : si on est capable de mettre un objet flash dans un upload ?
-> pour ces 2 classes de pbm, pas vraiment de solution. Comment s'en sortir -> tous les uploads vont sur un serveur séparé.

3/ contournement d'application
- Accès non sécurisé au serveur (ftp, mot de passe faible)
- injection de code (injection sql, injection code script php) - solution : module php à bannir ? utiliser les fonctions de l'api.

info contenue dans le dump de la base - avant dans drupal, on pouvait récupérer un token de connexion dans le dump !!! jusqu'à l'année derniere !!!
solution : ajouter un ?

problemes sur les entetes host (pour faire croire à un site qu'on est un autre site). IL FAUT

REGLER LE BASE URL DANS LE SETTING.PHP ! CA N'EST PAS OPTIONNEL - sinon on peut faire deviner à drupal une mauvaise base url.

les open redirects : impact reste à définir

SI QQ1 d'AUTRE VEUT NOTER LES NOTES - N'HESITEZ PAS A VOUS MANIFESTER LES 3 AUTRES - mon niveau de compréhension n'est pas le meilleur pour que je rapporte cette session