Traitement des incidents de sécurité dans le logiciel libre

Matthieu Herrb

CNRS-LAAS/X.Org

Envol, 22 octobre 2008

http://www.laas.fr/~matthieu/talks/envol08-incidents-secu.pdf

Agenda

Agenda

X.Org

- Projet qui développe le système de multi-fenêtrage X (The X Window System).
- Animé par la fondation X.Org
- Membres de la fondation : toute personne impliquée dans le projet (développeur, testeur, soutien,...) gratuit. Critères définis par les status.
- Sponsors de la fondation : entreprises qui donnent de l'argent.
- Budget : environ 40k$ annuels : 2 conférences, achat de matériel, vacation of code.
- Hébergé par l'Université de Portland (`www.freedesktop.org`) et le MIT (`www.x.org`).

Publication des correctifs de sécurité pour un Logiciel Libre

La transparence du logiciel libre peut poser des problèmes en cas de publication trop rapide d'un correctif pour un bug de sécurité :

- fournit suffisament d'infos pour les pirates pour exploiter la vulnérabilité...
- ...sans laisser le temps aux utilisateurs honnêtes de corriger pour se protéger !

Publication raisonnée

Principe :

- ne pas publier de suite les correctifs,
- utiliser un moyen non public pour informer les acteurs concernés,
- lorsque tout le monde est prêt, publier l'avis de sécurité et les correctifs ensemble.

Documenter tous les détails ?

- rend plus facile l'exploitation
- pas toujours facile à décrire précisément

Agenda

Structures existantes

MITRE CVE `http://cve.mitre.org/` base de données des les vulnérabilités (pas limité au logiciel libre). suivi des bugs d'une application entre toutes ses distributions

oss-security `http://oss-security.openwall.org/wiki/` resources sécurité pour projets Open Source.

oCERT Open Source CERT `http://www.ocert.org/`. Fournit des resources pour aider les projets Open Source à traiter leurs pbs de sécurité.

`vendor-sec@lst.de` liste non publique utilisée par les distributions de systèmes (libres et non-libre) pour coordoner la publication des avis.

Limites

- Difficile de garder la confidentialité - d'identifier les sources de fuites
- Grand nombre d'acteurs
- Retarder trop la publication d'un correctif officiel laisse les utilisateurs vulnérables en cas de re-découverte ou de fuite.
- Les bugs découverts via un exploit en circulation (Zero-day) doivent être corrigés au plus vite.

Le marché des vulnérabilités

Quelques sociétés spécialisées en sécurité vendent à leurs clients des informations “en avance” sur les vulnérabilités et les correctifs.

- achêtent les bug reports auprès des chercheurs,
- préparent un rapport (éventuellement un correctif),
- le diffusent à leurs clients,
- après quelques semaines :
  - contactent l’auteur initial,
  - acceptent la publication.

En pratique

- Créer un point de contact facilement identifiable pour signaler les pbs de sécurité de manière confidentielle. (par ex. : `security@project.org`)
- Se faire connaitre auprès de la communauté (wiki oss-sec,...) pour être tenu au courant des pbs qui peuvent affecter son code.
- En cas de pb, informer **vendor-sec** et **oCERT**.
- Négocier la date de publication avec eux.

Choix de la date de publication

- pas une veille de WE ou (pire) de pont
- pas le lundi matin (souvent occupé par d'autres choses)
- le mardi ou le jeudi sont des bons jours.
- tenir compte des fuseaux horaires. 14 :00 GMT est généralement considéré comme le moins pire des compromis.
- éviter le 3e mardi du mois (patch tuesday de Microsoft) déja assez occupé en général.

Ce qui est important

- ne pas se précipiter pour publier un bug-report ou corriger un bug si il peut avoir un impact de sécurité.
- discuter du pb avec des personnes de confiance uniquement.
- ne pas ignorer ou laisser traîner inutilement un problème.
- ne pas minimiser les conséquences possibles d'un bug
- reconnaître ses erreurs et en informer les autres rapidement.
- ne pas culpabiliser inutilement. "Shit happens".

Sécurité des structures

Assurer la confidentialité, l'authenticité et l'intégrité des échanges.
Repose en général sur PGP.

- diffuser sa clé publique :
  - pour permettre l'envoi de messages chiffrés au point de contact.
  - pour signer les avis et les correctifs
- Avoir sous la main les clés publiques des partenaires
- Ne jamais faire suivre en clair un message reçu chiffré (sauf autorisation expresse)
- considérer toutes les informations comme sensibles et les protéger comme telles (chiffrement sur disque, pas sur un poste en libre-service ou une clé usb qui circule, ...

CERTs traditionnels

Relation difficiles....

- grosses machines administratives manquant de réactivité
- manque de confiance mutuelle
- commettent autant d'erreurs que les "amateurs" du logiciel libre
- imposent souvent des contraintes de confidentialités déraisonnables.

Agenda

Exemple : X.Org

- adresse de contact `security@lists.x.org`
- regroupe des développeurs experts en sécurité de X et quelques distributeurs majeurs
- membre de `vendor-sec`
- bugs privés dans bugzilla : visibles des seuls utilisateurs autorisés

Exemple : CVE-2009-666

6/3/2009 jolibug@gmail.com envoie un message chiffré et signé à security@acme.org signalant un débordemenent de buffer dans le code de CaFaitTout version 3.33.

7/3/2009 joe@acme.org confirme l'existence du problème et analyse qu'il à été introduit dans la version 2.45. Il répond à jolibug pour accuser réception du pb et lui demande si un CVE-Id a été attribué ou une date de publication fixée. Il propose également un premier correctif.

7/3/2009 jolibug répond que non, mais que. à condtion que le délai n'éxède pas 2 mois, il est prêt à différer la publication du problème.

7/3/2009 Sachant que CaFaitTout est inclus dans de nombreuses distributions Linux, joe contacte vendor-sec@lst.de, en explicant le problème, listant les versions vulnérables. Il demande l'attribution d'un CVE-ID et propose le 30/4/209 14 :00 GMT comme date de publication.

8/3/2009 Paul.Robert@BlueCap.com répond à vendor-sec qu'il a attribué l'ID CVE-2009-666 à la vulnérabilité et signale que le 30/4 n'est pas une bonne date pour la publi, veille d'un long WE.

10/3/2009 joe reconnait ne pas avoir regarder le calendrier de près et propose alors d'avancer la publication au 21/4.

11/3/2009 plusieurs membres de vendor-sec acquièsent.

6/4/2009	eric@nobugs.com signale à vendor-sec et à joe que son correctif est incomplet : un problème similaire se produit dans une autre fonction.
7/4/2009	nouvelle version du correctif diffusée sur vendor-sec
15/4/2009	joe diffuse aux membres de vendor-sec un brouillon de son avis de sécurité et la version définitive du patch.
21/4/2009	14 :00 GMT envoi de l'avis de sécurité sur la mailing list du projet CaFaitTout annoncant la version 3.34. joligug@gmail.com envoie son avis à bugtraq et full-disclosure. Peu après, Debian, Gentoo, RedHat, Fedora Suse, Mandriva, OpenBSD et FreeBSD publient des mises à jour de leurs paquets 2.33.
22/4/2009	Le CERT-A et le CERT-Renater rediffusent l'info sous forme synthétique.

Agenda

Conclusion

- Un traitement responsable des bugs de sécurité est indispensable.
- Tant pis pour la “full disclosure”.
- Ne pas rester isolé
- Éviter de publier du code passoire. Se faire aider avant la publication.
- Prévoir l’infrastructure pour le traitements des incidents dès la mise à disposition initiale du projet

Questions ?