



Projet fusil



- Fuzzing
- Programme `target.py`
- Programme `http_server.py`
- Résultats

Victor Stinner – SSTIC 2007

Fuzzing



- Analyse d'un système considéré comme une boîte noire
- Recherche des points d'entrée
- Partir de données valides en ajoutant des erreurs

target.py



- Détection des fichiers accédés et des variables d'environnement avec ltrace
- Fuzzing sur chaque point d'entrée
- Sondes pour surveiller le comportement du programme

mangle.py



- Opérations :
 - remplacer un octet par un octet aléatoire
 - valeurs spéciales (0x00, 0x7F, 0xFF, ...)
 - incrémenter/décrémenter un octet
 - insérer/supprimer un octet
- Problème : choisir les bons paramètres

Vulnérabilités



- **glibc (printf)**
- **LibEXIF**
- **ClamAV (OLE2 = Word)**
- **FreeType (TTF)**
- **gettext, rpm, poppler (pdf), binutils, vim, ImageMagick, xterm, dpkg-query, ...**

Réponses



- Trois types de réponses à mes rapports de bug
- *On s'en fout* (ex: ImageMagick, gettext)
- *Ce bug est (sera) corrigé* (ex: glibc, ClamAV)
- *Très intéressant !* (ex: rpm, FreeType2)

Pan !



*** Démonstration de target.py ***

http_server.py



- Serveur HTTP générant des fichiers fuzzés
- Formats testés : JPEG, PNG, GIF, ...
- Stockage des derniers fichiers générés

http_server.py



- Cinq bugs trouvés dans Flash9 (swf)
- Plantage de Konqueror : image TIFF

Loop# 15 (files: 150). Time: 45.2 sec. Last load duration was 3.24 sec.



Fuzz: 10 files (image/jpeg). Reload <http://localhost:8080/> in 1 seconds.

Conclusion



- Succès rapide sur les applications testées
- Les fichiers sont souvent négligés
- Nombreuses évolutions possibles
- **<http://fusil.hachoir.org/>** (licence GPL)

Questions



¿ Questions ?