Analyse différentielle de binaires

Kostya Kortchinsky
Responsable du CERT RENATER
kostya.kortchinsky@renater.fr

Introduction

- Identification d'une vulnérabilité suite à la publication d'un correctif
  - Les détails d'une vulnérabilité corrigée peuvent ne pas être communiqués
    - MS05-027 : SMB
  - Un éditeur peut être tenté de corriger des failles silencieusement
    - MS04-007 : ASN.1 (Kill Bill)
- Une comparaison manuelle serait un travail titanesque
- Automatisation de l'analyse et *visualisation* des différences grâce à des graphes
  - Public : BinDiff de Sabre Security
  - Privé : BindView, NSFocus, iDEFENSE, Tenable, RENATER, ...

Concept

- Les méthodes traditionnelles se révèlent inefficaces
  - Comparaison octet à octet
  - Comparaison du texte désassemblé
- Représentation des unités logiques basiques du binaire sous forme de graphes
  - Les fonctions sont des graphes d'instructions
    - Ou de blocs d'instructions consécutives
  - Le binaire est un graphe de fonction
- Le problème est alors celui d'un **isomorphisme** de graphes
  - Pas non plus forcément évident ...

Signatures

- Afin de mettre en avant des similarités entre les graphes qui seront les points fixes de l'isomorphisme
- Signatures structurelles
  - Fondées sur les caractéristiques du graphe
    - Peu dépendantes du langage machine
- Signatures conjoncturelles
  - Fondées sur le contenu des blocs basiques constituant le graphe
    - Très dépendantes du langage machine
- Chaque signature a ses avantages et ses inconvénients

Visualisation

- Il est plus facile pour un humain (normalement constitué) d'étudier les différences entre deux organigrammes colorés qu'entre des lignes d'assembleur
  - La transition depuis la description choisie est immédiate
- Représentation textuelle possible grâce au format GDL (Graph Description Language)
- Un outil supportant ce format est inclus dans IDA Pro : WinGraph32

Applications (1/2)

- **BinDiff**
  - Plugin pour **IDA** Pro 4.8 pour processeurs x86, MIPS, SPARC, PowerPC
  - Analyse différentielle de deux binaires
    - Analyse de fonctions par blocs
    - Génération de signatures pour chaque fonction
      - Nombre de blocs, nombre d'arcs, nombre de calls
    - Mise en correspondance des fonctions des deux binaires
    - Mise en évidence des fonctions modifiées
    - Visualisation des différences grâce aux graphes des fonctions
      - Jusqu'au niveau de l'instruction
  - Algorithme de mise en correspondance
    - Nom des fonctions
    - Signatures
    - Arbre d'appels aux fonctions
    - Références aux chaînes de caractères

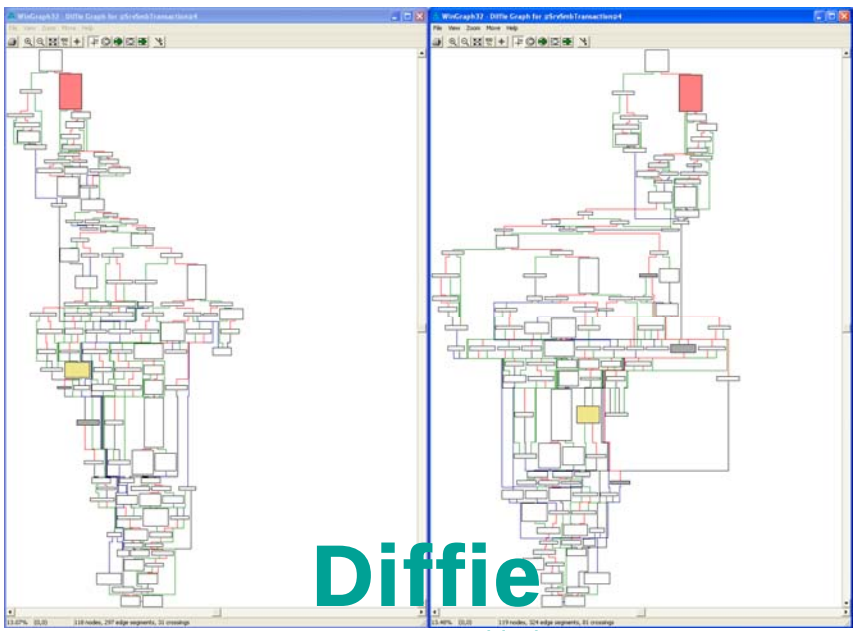
Applications (2/2)

- **Diffie**
  - Plugin pour **IDA** Pro 4.8 pour processeur x86
  - Analyse différentielle de deux binaires
    - Analyse de fonctions par blocs
    - Génération de signatures type **CRC32** pour chaque bloc et pour chaque fonction
    - Mise en correspondance des fonctions des deux binaires
    - Mise en évidence des fonctions modifiées
    - Visualisation des différences grâce aux graphes des fonctions
  - Algorithme de mise en correspondance
    - Nom des fonctions
    - Signatures
    - Position dans le binaire
    - Arbre d'appels aux fonctions

Utilisations

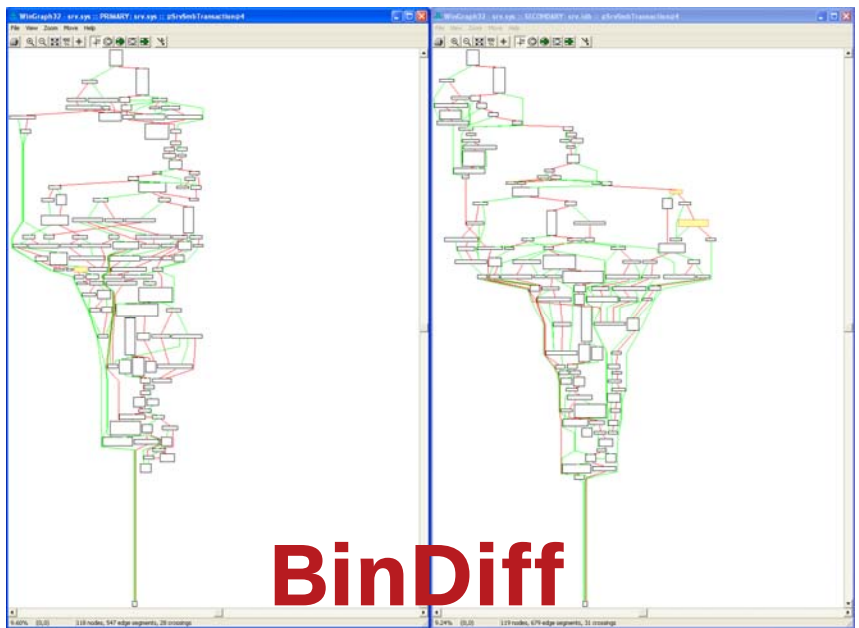
- Qualifier une vulnérabilité suite à publication du correctif
  - Déterminer le type de la vulnérabilité
    - Les bulletins sont rarement explicites
  - Déterminer l'impact potentiel
    - Facilité et fiabilité de l'exploitation
      - Dépendance à la régionalisation du système d'exploitation
      - À sa version (service pack)
- Trouver de nouvelles vulnérabilités en analysant les différences entre versions
  - Vulnérabilités corrigées dans Windows XP SP2
- Comparer différentes versions de « malware »
- Déterminer les similarités de code potentielles (vol ?)
  - Cas PearPC vs CherryOS

Démonstrations



Diffie

par Kostya Kortchinsky



BinDiff

par SABRE Security

Conclusion

- Les correctifs binaires sont souvent aussi explicites que des « diff » de code source
- L'implémentation d'un exploit dans les milieux compétents est une question d'heures
  - Les informations disponibles au public sont loin de refléter la réalité
    - Retard de plusieurs jours, voire mois (Kill Bill)
    - Qualité des exploits bien en deçà de ce qu'il est possible de faire
- Les résultats des analyses différentielles restent souvent privés
  - Trouver des 0days dans d'autres applications grâce à une faille similaire est commun

Littérature

- Papiers
  - Graph-based comparison of Executable Objects par Thomas Dullien et Rolf Rolles, SABRE Security
  - Comparing binaries with graph isomorphisms par Todd Sabin, Bindview RAZOR Team
- Présentations
  - Compare, Port, Navigate par Thomas Dullien et Rolf Rolles, SABRE Security
  - Binary Comparison of Security Patch par Hume, NSFocus
  - Structural Signature and Signature's Structure par Funnywei, Xfocus TEAM

Questions ?

Remerciements :

Dave, Halvar, Gera, Security Labs, Team Rstack, Sécurité.org, Microsoft