

DE LA RECHERCHE À L'INDUSTRIE



Sécurité applicative

Forensics – Extraction des codes malveillants

Commissariat à l'Énergie Atomique et
aux Énergies Alternatives | Camille MOUGEY – Mathieu BLANC

8 janvier 2019

Objectifs

- Etre prêt en cas d'intrusion / compromission
 - Préparation à la gestion d'incidents de sécurité
 - CERT : Réaction face aux attaques
 - Organisation des ressources humaines
- Mettre en place des moyens de remontée d'incidents
 - Identifier les menaces
 - Surveillance / Détection
- Savoir analyser un système compromis
 - La démarche
 - Les outils d'analyse

- 1 Aspects organisationnels**
 - Gestion des incidents de sécurité
 - Cycle de gestion des incidents
- 2 Les menaces**
 - DoS
 - Malwares
 - Accès frauduleux
- 3 Démarche d'analyse forensique**
 - Analyse live / offline
 - Collecte des preuves

- 1 Aspects organisationnels**
 - Gestion des incidents de sécurité
 - Cycle de gestion des incidents
- 2 Les menaces**
 - DoS
 - Malwares
 - Accès frauduleux
- 3 Démarche d'analyse forensique**
 - Analyse live / offline
 - Collecte des preuves

- Organiser ses capacités de réponse
 - définir des politiques et procédures
 - structurer une "response team"
- Gérer les incidents
 - de la préparation initiale aux leçons tirées *a posteriori*
- Gestion spécifique des menaces
 - déni de services : privation ou saturation de ressources
 - code malicieux : virus, ver, trojan ... infection
 - accès non autorisé : contournement des permissions mises en place
- Mener une analyse post-intrusion
 - Les rootkits
 - Préservation des données
 - Analyse « live » et « offline »

- Menace : potentialité d'attaque, décrite par :
 - un élément menaçant (naturel, environnemental ou humain)
 - une typologie (portée, contexte, conditions, ...)
- Événement de sécurité (EdS)
 - tout événement portant atteinte à la sécurité du SI ou susceptible de lui porter atteinte
- Alarme
 - EdSs déclenchant un signal automatiquement en fonction d'une gravité pré-établie
- Incident
 - EdSs qualifiés, demandant une intervention pour rétablir la situation
- Attaque
 - menace "passée à exécution", détectée à partir de l'analyse des EdSs et des incidents

■ Politique de sécurité

- objectifs : présentation du contexte d'application
- périmètre : à qui, à quoi et quand l'appliquer
- structure : organisation (rôles, responsabilités et autorités)
- classification des incidents en fonction de leur gravité
- moyens d'évaluation
- reporting et contacts

■ Procédures

- escalade / redescente du niveau d'alerte
- information des contacts externes
- check-lists et rapports d'incidents pre-formatés

Charte de sécurité

- La « partie visible » de la politique de sécurité
- Composants
 - accès au SI : soumis à autorisation
 - confidentialité : les données sont privées
 - règles d'utilisation : décrit qui peut faire quoi
 - considérations légales : lien avec les lois en vigueur
- Objectifs
 - responsabiliser les utilisateurs et administrateurs
 - permettre d'entamer des poursuites si besoin

- centralisée : une équipe pour toute l'organisation
 - petites organisations, ou organisations avec informatique centralisée
- distribuée : plusieurs équipes, chacune dédiée à un secteur logique ou physique
 - grande organisation, mais les équipes doivent appartenir à une unique entité
 - communication permanente entre les équipes
- coordonnée : une équipe principale guide et conseille des équipes locales
 - organisation très hiérarchique

- Alerte de sécurité
 - annonce de vulnérabilités, symptômes d'attaques, ...
- Evaluation de vulnérabilité
 - examens des vulnérabilités réseau, système ou applicative afin d'évaluer le risque
- Détection d'intrusion
 - analyses des événements à la recherche d'intrusion
- Formation et sensibilisation
 - le maillon faible reste l'Homme ...
- Veille technique et scientifique
 - la sécurité évolue très rapidement

Remonter les incidents

■ 1. Hot line

- reçoit une alerte (utilisateurs mécontents, ...)
- qualification initiale de l'incident
- si intrusion possible, transmettre aux experts et au responsable technique

■ 2. Cellule d'experts

- re-qualification de l'incident
- si incident de sécurité avéré, resp. technique prévient le RSSI
- puis le RSSI prévient le comité de direction

■ 3. Comité de direction

- évaluation de l'incident en terme financier, juridique et médiatique
- élaboration de la stratégie globale de réponse

■ Responsables

- opération : adaptation de la chaîne de production à l'incident
- financier : mise à disposition de moyens financiers
(prestations externes, remplacement de matériel, ...)
- communication : si décidée par le comité de direction
- juridique : alerte des services appropriés

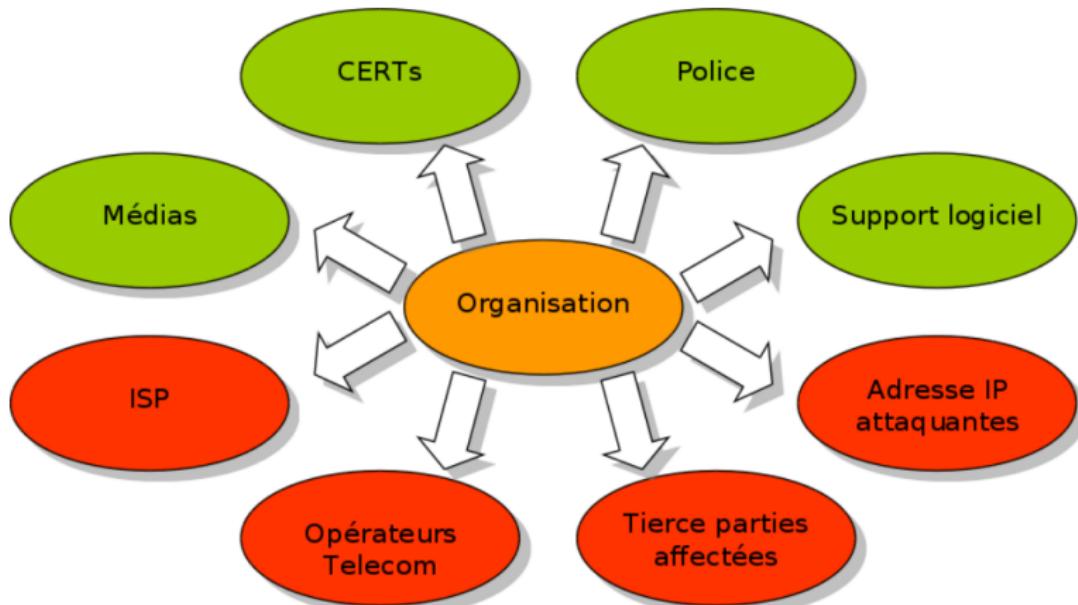
■ Cellule d'experts

- définition et application du plan de remise en état
- communication permanente avec le comité de direction

■ Hot line

- prévient les utilisateurs, en fonction de la politique de communication interne

Relations externes



Relations externes

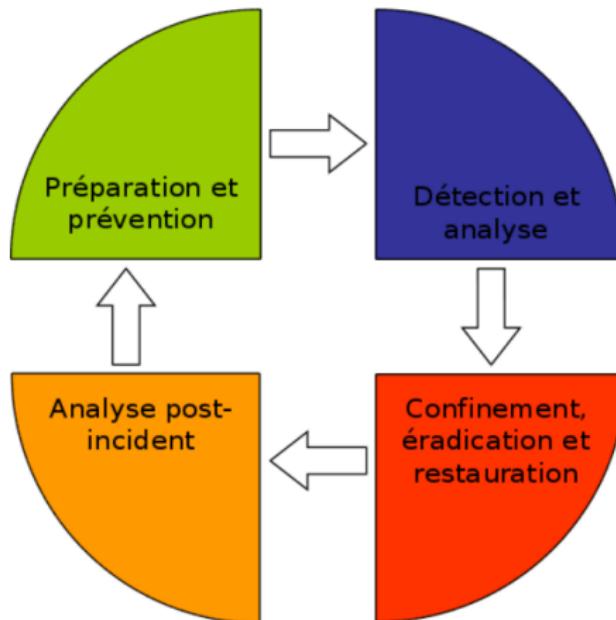
■ Relations préservées de l'attaque

- média : décrypter l'attaque avant que la communication ne devienne elle-même un vecteur
- response team : partage d'informations (CERT / CC)
- justice : contacter la police (BEFTI, DST, ...)
- organisations type CERT (Administration, RENATER, ...)
- vendeur : origine de la faille

■ Relations concernées par l'attaque

- FAI : conservation des logs + dommages collatéraux
- télécoms : surcharge des lignes par utilisateurs
- relations externes : perte de documents confidentiels
- origine : peut être victime, complice ou l'attaquant

Cycle de gestion



- Ressources humaines dédiées (CERT)
- Outils matériels et logiciels
 - Environnements d'analyse déconnectés
 - Prévoir toutes les connectiques, bloqueurs d'écriture
- Ressources liées au système d'information
 - Documentations, hash des fichiers sensibles, diagrammes des réseaux, patches, backups
- Prise en compte des incidents passés
 - Base d'intelligence, indicateurs de compromissions (IOCs)
 - Base de suivi d'incidents
 - Plateformes d'échange (MISP...)

- Gestion des correctifs
- Sécurité système et réseau classique
 - Guide de configuration et durcissement des systèmes
 - Interdire tout ce qui n'est pas explicitement autorisé
- Détection de codes malveillants
 - Mise à jours des anti-virus, (reverse) proxies, et autres
 - Sondes de détection
- Sensibilisation et formations des utilisateurs
 - Information sur la politique de sécurité, responsabilisation

■ Sources de détection d'incidents

- logiciels : {H,N}IDS, audit, antivirus
- logs : systèmes, applicatifs, réseau, honeypots
- sources ouvertes : listes de diffusion, organismes (CERT)
- personnes : utilisateurs, administrateurs, "contacts"

■ Types de détection d'incidents

- a priori : signe qu'un incident peut survenir
annonce d'une faille, audit, menace identifiée
- a posteriori : signe qu'un incident a pu se produire
crash d'un serveur, alerte d'un AV, postmaster submergé
par des mails d'erreurs

- connaître son environnement
 - identifier ses caractéristiques (audit système et réseau)
 - connaître le comportement normal
- gérer ses logs
 - sauvegarde des logs sur une période de temps fixée
 - centralisation et corrélation d'événements
- organiser ses connaissances
 - mise en place d'une base de connaissances
 - création de fiches d'incidents

Critères de stratégie :

- choix des priorités
 - dommages, preuves, disponibilité, temps / coût, efficacité
 - disponibilité du service / gravité de l'incident
- préservation des preuves vs. confinement
 - la moindre action peut détruire des traces
 - le pirate va tenter d'effacer ses traces s'il se croit repéré

Identifier le coupable :

- juste un bonus, pas une priorité
 - étape souvent longue
 - pas forcément utile
- remonter la piste
 - scan (ports/vuln), moteur de recherche, bases d'incidents, surveillance des canaux de communication, hack back, ...

Détection et Analyse :

- Déterminer les priorités en fonction de l'impact
 - Identifier les ressources affectées, prédire celles qui le seront
 - Estimer les dégâts et leurs conséquences
 - Attribuer les priorités en fonction des résultats précédents
- Rapporter l'incident (interne/ externe)

Confinement, éradication et restauration :

- Récupérer et protéger les preuves
- Confiner l'incident

Eradication de l'incident

- Identifier et corriger les vulnérabilités exploitées
 - Nettoyer les matériels et logiciels affectés
- ## Restauration du système de confiance
- Retour à un état opérationnel
 - Vérification du retour à la normale
 - Si besoin, mise en place d'une surveillance spéciale

Analyse post-incident :

- Edition et archivage d'un rapport
- Réunion de debriefing

1 Aspects organisationnels

- Gestion des incidents de sécurité
- Cycle de gestion des incidents

2 Les menaces

- DoS
- Malwares
- Accès frauduleux

3 Démarche d'analyse forensique

- Analyse live / offline
- Collecte des preuves

- Une menace est :
 - une faiblesse résiduelle du système identifiée
 - à facteur interne (erreurs de configuration, patches inapplicables ...) ou externe (risque de DoS, recherche de vulnérabilités ...)
- Afin de gérer l'occurrence d'un incident de sécurité, on identifie pour chaque type de menace :
 - nature (définition, exemples)
 - actions préventives
 - signes a priori et a posteriori de sa réalisation
 - preuves à conserver
 - confinement, éradication, restauration

Définition :

- Un DoS est une attaque qui vise à mettre en défaut la disponibilité de ressources, de services
 - par épuisement
 - par provocation d'une erreur

Exemples :

- épuiser les ressources d'un réseau
 - synflood, smurf, réflexion (DNS, echo, isakmp, ...)
- épuiser les ressources d'un système
 - accaparer toute la mémoire ou le CPU, *while(1), fork()*
- épuiser les ressources d'une application
 - attaque par complexité sur les tables de routage
- provoquer une erreur dans une application
 - introduction de données forgées

■ Préparation

- refléchir à son infrastructure réseau
- mettre en place un monitoring des ressources
- protéger l'accès aux informations sensibles
- contrôle les données d'entrée des services

■ Prévention

- systèmes et services redondants
- interdire tout ce qui n'est pas explicitement autorisé
- limiter l'utilisation du réseau
 - QoS : limiter la bande passante
 - limiter le nombre de connexions simultanées
- mise en place de reverse proxies

- *Signes a priori*

- activité anormale en guise de reconnaissance
 - découverte d'une nouvelle attaque / outil

- *Signes a posteriori*

- DoS contre un hôte / réseau
 - ralentissement, pertes de connexions, trafic asymétrique
 - DoS contre un système
 - alertes HIDS, ralentissement, reboots intempestifs
 - DoS contre une application
 - alertes {H,N}IDS, ralentissement, logs atypiques

- Préservation des preuves

- identifier l'origine du trafic illégitime
 - déterminer comment les zombies ont été compromis
 - éplucher les logs surabondants générés par le DoS

- Confinement, éradication, restauration

- basculer sur le "plan B"
 - patcher la faille exploitée
 - rerouting / filtrage du trafic (local + ISP)
 - hack back sur les zombies
 - attendre que ça passe ...

Définition :

- malicious software (malware) : programme capable de perpétrer une action interdite sur un SI

Exemples :

- Malware simple
 - Bombe logique
 - Cheval de Troie
- Malware auto-répliquant
 - Virus
 - Vers
 - Cheval de Troie évolué

■ Préparation

- sensibiliser ses utilisateurs
 - réflexe du clic sur les pièces jointes
 - problème des portables qui entrent et sortent de l'entreprise
- s'informer avec les bulletins des éditeurs anti-virus
- configurer des HIDS sur les systèmes critiques

■ Prévention

- installation d'anti-virus – à jour – partout
- Network Access Control
- restreindre l'accord de droits admin aux utilisateurs
- configuration rigoureuse des clients mails / web
- filtrage applicatif, passerelle anti-virale, proxy web

Malwares : acte 2

■ Signes *a priori*

- alerte prévenant d'un malware sur un logiciel utilisé
- malware subitement détecté

■ Signes *a posteriori*

- e-mail contaminant
 - nb de mails croissant, système instable, ralentissement, fichiers corrompus ou détruits
- ver sur un réseau
 - trafic réseau intense, système instable, service inaccessible
- cheval de Troie
 - processus anormal, port inhabituel ouvert, alertes NIDS
- code mobile
 - dialog boxes intempestives, connexions suspectes

- Préservation des preuves
 - pratiquement aucune chance d'identifier l'auteur
- Confinement, éradication, restauration
 - identifier et isoler les hôtes contaminés
port scans, honeypots, AV, ...
 - fermer la source d'infection (mail / web)
 - coupure de l'accès à Internet
 - tester et nettoyer les postes potentiellement infectés

Accès frauduleux

Définition :

- un accès frauduleux se produit lorsqu'un utilisateur acquiert un niveau de privilège non prévu dans la politique de sécurité

Exemples :

- local / remote root exploit
 - le pire des cas
- récupérer / casser des mots de passe
- dumper une base de données
- défaçonner un serveur web

Accès frauduleux : acte 1

■ Préparation

- informer sur les risques économiques / organisation, juridiques / utilisateurs
- déployer des {H,N}IDS, gérer et analyser les logs

■ Prévention

- sécurité réseau
 - architecture, sécuriser / contrôler les accès remote (vpn)
- sécurité système
 - audits de vuln réguliers, minimisation de privilèges
- authentification
 - password policy, choix des méthodes (fortes / faibles)
- sécurité physique

Accès frauduleux : acte 2

■ Signes *a priori*

- annonce d'une faille dans un logiciel utilisé
- tentatives de social engineering
- nombreux échecs d'authentification

■ Signes *a posteriori*

- accès root / admin
 - changement de configuration, fichiers altérés / créés,
 - utilisation anormale (idle, hors horaire)
- modification de données (warez, defacement)
 - alertes IDS, utilisation anormale de ressources,
 - fichiers altérés / créés
- utilisation frauduleuse d'un compte "normal"
 - tentatives d'accès à des fichiers critiques,
 - utilisation anormale (idle, hors horaire)

Accès frauduleux : acte 3

■ Préservation des preuves

- créer une image complète du système compromis
copie bit à bit par un système homologué
- sauvegardes des logs (système + environnement)

■ Confinement, éradication, restauration

- isoler le système
- désactiver le service / compte abusé
- bloquer / remonter la route prise par l'intrus
souvent le type d'intrusion le plus difficile à reconstituer ...

Sommaire

1 Aspects organisationnels

- Gestion des incidents de sécurité
- Cycle de gestion des incidents

2 Les menaces

- DoS
- Malwares
- Accès frauduleux

3 Démarche d'analyse forensique

- Analyse live / offline
- Collecte des preuves

La démarche

- L'analyse forensique a pour but :
 - de mettre en évidence la technique utilisée par l'attaquant
 - d'identifier les faiblesses exploitées dans le système
 - de relever exhaustivement les données ajoutées, modifiées ou supprimées
 - de préserver l'ensemble des traces liées à l'incident pour preuve
- On parle aussi d'investigation, d'autopsie
- Le problème de la confiance
 - Confiance dans le système attaqué (connaissance du système, de ses faiblesses)
 - Confiance dans les outils d'investigation (intérêts stratégiques, politiques)
 - Confiance dans l'investigateur (corruption)

Reflections on trusting trust (Thompson 1984)

- ajout d'un backdoor dans /bin/login
 - accès root pour tous les systèmes avec ce binaire
- les sources login.c sont présentes sur le système
 - tout le monde peut remarquer la backdoor dans les sources
 - Thompson remet le login.c propre
- l'admin peut recompiler login.c → /bin/login 'propre'
 - Thompson modifie le compilateur : s'il compile login.c, ajout de la backdoor
- les sources du compilateur sont présentes sur le système
 - tout le monde peut remarquer la backdoor dans les sources
 - Thompson remet le compilateur.c propre
- compilateur C écrit ... en C
 - le binaire du compilateur reconnaît ses propres sources,
et ajoute sa backdoor pour login.c

- Système compromis : l'intrus peut
 - utiliser les ressources
mémoire, disque, bande passante, ...
 - récupérer des données
keylogger, locate mp3 avi visa
 - rester invisible sur le système
modifier le comportement du système pour se cacher
- Système compromis : l'admin peut
 - détecter les fichiers / processus modifiés
 - restaurer l'intégrité du système ...
- Mais peut-on encore faire confiance au système ?

- Intrus : modifier les binaires

- changer le comportement normal des commandes
 - ps pour cacher les processus de l'intrus
 - netstat pour cacher les connexions de l'intrus

- Du côté de l'admin

- si une seule commande est oubliée, l'intrus est vu
 - détection fondée sur des fonctions de hashage
 - base de référence des hash sur un support read-only

```
$ md5sum /lrk5/ifconfig 086394958255553f6f38684dad97869e
$ md5sum 'which ifconfig' f06cf5241da897237245114045368267
```

- De l'importance de créer une base d'empreintes ...

- sauf si le programme de vérification est compromis!

- Intrus : modifier les bibliothèques
 - une bib. dynamique affecte plusieurs programmes
 - moins de modifications, plus de discrétion

```
$ ldd 'which uptime' 'which ps'  
/usr/bin/uptime:  
libproc.so.3.2.8 => /lib/libproc.so.3.2.8 (0x40025000)  
libc.so.6 => /lib/libc.so.6 (0x40032000)  
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)  
  
/bin/ps:  
libproc.so.3.2.8 => /lib/libproc.so.3.2.8 (0x40025000)  
libc.so.6 => /lib/libc.so.6 (0x40032000)  
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

- Du côté de l'admin
 - emergency kit contenant des binaires statiques

- Intrus : modifier le noyau
 - difficile de patcher tous les binaires et bib. dyn.
 - attaquer la seule ressource partagée : le noyau
- L'espace noyau, un nouvel El Dorado
 - nouvelle génération de root-kits
 - intrus plus puissant que root / admin
 - contrôle complet sur le user space
 - sniffer avant le firewall
 - ajouter des kernel threads invisibles

- Autres magouilles
 - patcher les headers des fonctions (`jmp <fake func>`)
 - injection de kernel / user thread (héritage des droits du processus)
- Du côté de l'admin
 - outils de détection : agissent en userland
AIDE, Tripwire, chkrootkit, rk hunter, ...
 - nécessité de vérifier l'intégrité en espace noyau
vérification d'adresses, d'instructions, ...
 - ou de ruser ...
mesure du temps, vérifier `/proc/<pid>`

Objectifs :

- récupérer un maximum de preuves de multiples sources
 - disque dur, mémoire, réseau, ...
- préserver au max. ces preuves dans leur état initial
 - problème de la légalité de la preuve informatique
- reconstruire la succession d'événements
 - comprendre et corriger l'incident

Remarques :

- chaque incident est différent du précédent
 - en supposant que l'incident précédent a été corrigé
- partir avec aucun *a priori*
 - faille, outils, liens avec d'autres incidents, ...

Principes de précaution :

- ne pas se presser
 - prendre le temps de réfléchir à chaque information révélée
- enregistrer scrupuleusement chaque action
 - conservation des traces, évaluation des conséquences
 - intérêt du « logged shell »
- préparer la salle d'opération
 - création d'un toolkit adapté
 - configuration d'un serveur pour recueillir les preuves

- 1. Retirer le bruit
 - retirer les IP uniques, les machines connues, ...
- 2. Extraire les flux significatifs
 - sortants : agrégation par sous réseau (scans), par protocoles, ...
 - entrants : flux longs vers des ports non classiques, ...
- 3. Construire la timeline
 - trier les flux par date
- 4. Corréler les informations
 - corréler les informations avec celles du système

■ Comment réaliser l'analyse ?

- outils standard comme tcpdump ou wireshark : difficile de travailler au niveau paquet
- travailler au niveau connexions ou flux avec wireshark ou bro
- logs réseau (firewalls, IDS, ...)

■ Où les flux réseau viennent à notre aide

- flow : ensemble de paquets avec des caractéristiques communes
L3 / L4 information, temps / durée, nb de paquets / octets
- conçu pour la comptabilité et facturation
prévu pour travailler sur de gros volumes de données
- utilisé depuis des années par les admins
protocoles et outils matures, outils type netflow, argus

- Énumérer les protocoles

- regarder les protocoles niveau 3 et 4
 - vérifier protocoles étranges / covert channels
 - GRE tunnel sur IPv6, raw IP, ...

- Challenge : identifier les sessions TCP

- Essayer d'identifier les volumes importants
 - Chercher les ports destination qui sortent de l'ordinaire

Première analyse en direct sur le système compromis (pas toujours possible)

- Données volatiles
 - Données relatives au système en fonctionnement
 - A sauver sur un médium indépendant (USB, net, ...)
- Cahier d'opérations
 - Date, heure, ligne de commande, lieu de stockage des résultats de la commande, empreinte de la sortie
- Identifier le système
 - Informations de configuration
 - stratégie d'audit, de MdP, services démarrés, prog. installés, uptime
 - Nom, version, date, config réseau, routage
 - Windows : hostname, ver, date /T, ipconfig, route print
 - Unix : uname -a, date, ifconfig, netstat -arn

■ Activités

- Confirmer / infirmer la présence de l'intrus
- Identifier les ressources utilisées par l'intrus
- Utilisateurs, processus, sockets, table ARP
 - Windows : tasklist, netstat -an(o|b), arp -a
 - Unix : w, ps auxwww, netstat -atupn, arp -an

■ Historique des opérations

- Retrouver les opérations post-intrusion
- Identifier l'origine de la compromission
- Connexions locales, distantes, échecs
- Derniers fichiers accédés, dernières connexions / commandes

- Tracer les processus

- `ptrace()` : accès en debug aux processus
`strace`, `ltrace`, `ktrace`, `truss`, dump mémoire
- environnement processus : accessible depuis `/proc/<pid>/`
`/proc/<pid>/mem` : mémoire du processus
`/proc/<pid>/cmdline` : voir les arguments
`/proc/<pid>/fd/` : les fichiers utilisés
- mémoire noyau : accessible depuis `/dev/kmem`

- Collecte automatisée

- Outils à déployer préalablement
- Par exemple Google Rapid Response (GRR), Mozilla Investigator (MIG)

Extraction des données

■ Extraire tous les fichiers

- copie bit à bit des partitions de données

local : dd if=/dev/hda of=/dev/hdb

distant : dd if=/dev/hda | nc server 1703

- extraction de la RAM

Linux : LiME ou fmem (pas possible de lire /dev/mem)

Windows : win32dd.exe

■ Analyse du swap

- sauvegarder les partitions de swap

Windows : pagefile.sys, hiberfil.sys

Manipuler les données sauvegardées :

- connecter le disque dur
 - sur un système sain
 - en read only
 - mieux : avec un mécanisme de blocage d'écriture hardware

Objectifs :

- rechercher les fichiers modifiés ou effacés
 - opérations fortement dépendantes du système de fichiers
FAT, NTFS, ext2/3, jfs, reiserfs, ...
- en extraire l'information
 - liée à la compromission, liée à l'intrus

- Rechercher l'information dans chaque bit
 - externe : taille, bib. dynamiques, header
 - chaînes : textes contenus dans le fichiers
 - sections : .rodata, .data, .init, .fini, ...
 - symboles : symboles si binaires non strippés
- L'extraction de ces infos peut être automatisée
- Problèmes
 - extraction manuelle pour les binaires "protégés"
 - analyse des résultats nécessite un expert

- Exécuter les codes malveillants dans un environnement supervisé
 - Machines virtuelles jetables (cuckoo, malwr.com)
 - Sandbox d'appels système (systrace)
 - Environnement d'analyse émulé (miasm...)
- Problèmes de l'analyse live
 - Dangereux!
 - Comportement du malware différent suivant l'environnement

- Outils commerciaux, reconnus par les tribunaux
 - Guidance Software Encase Forensic (Encase)
 - Access Data Forensic Toolkit (FTK)
 - Intègrent des méthodes génériques de recherches de données sur des images de disques durs
 - Orientés recherche de preuves, peu adaptés à la reconstitution d'une intrusion
- Outils open-source
 - SleuthKit
 - Autopsy (front-end pour SleuthKit)
 - Distributions Kali, System Rescue CD
 - Pas d'orientation particulière, mais ces outils requièrent une grande expérience pour donner de bons résultats

Conclusion

- La gestion des incidents requiert
 - une équipe technique hyper-compétente
haute expertise dans plusieurs domaines
 - une organisation irréprochable
pas (trop) de place à l'improvisation, le temps est critique
 - une anticipation de chaque instant
étant donné qu'on ne sait jamais ce qui va arriver
- Analyse post-intrusion
 - demande de la méthode et de la rigueur
 - demande une très bonne connaissance du système
 - course à l'armement
 - processus : rootkit vs. détection
 - fichiers : forensics vs. anti-forensics
 - Mais que faire des résultats ?

Références

■ Cellules CERT en France

- CERTA (SGDN/DCSSI)
<http://www.certa.ssi.gouv.fr/>
- CERT RENATER
<http://www.renater.fr/>
- CERT-IST
<http://www.cert-ist.com/>

■ Articles Wikipédia

- Informatique Légale
- Computer Forensics

Références

■ Sites spécialisés

- Forensics Wiki
<http://www.forensicswiki.org/>
- Sleuth Kit
<http://www.sleuthkit.org/>
- Encase
<http://www.guidancesoftware.com/>
- FTK
<http://www.accessdata.com/>

Commissariat à l'énergie atomique et aux énergies alternatives
Centre de Bruyères-le-Châtel | 91297 Arpajon Cedex
T. +33 (0)1 69 26 40 00 | F. +33 (0)1 69 26 40 00
Établissement public à caractère industriel et commercial
RCS Paris B 775 685 019

CEA DAM
DSSI
CTSI