

Chapitre 4

IDS/IPS

Un système de détection d'intrusion (ou IDS : Intrusion Detection System) est un mécanisme destiné à repérer des activités anormales ou suspectes sur la cible analysée (un réseau ou un hôte). Il permet ainsi d'avoir une action de prévention sur les risques d'intrusion.
(source : wikipédia).

Un IDS (Système de Détection d'intrusions) permet de :

- Surveiller
- Contrôler
- Détecter

Le système de détection d'intrusion est en voie de devenir un composant critique d'une architecture de sécurité informatique

- **Faux positif** : une alerte provenant d'un IDS, mais qui ne correspond pas à une attaque réelle.
- **Faux négatif** : une intrusion réelle qui n'a pas été détectée par l'IDS

Les différents types d'IDS

Il existe donc différents types d'IDS :

- Les systèmes de détection d'intrusions (IDS)
- Les systèmes de détection d'intrusions "réseaux" (NIDS)
- Les systèmes de détection d'intrusions de type hôte (HIDS)
- Les systèmes de détection d'intrusions hybrides
- Les systèmes de prévention d'intrusions (IPS)
- Les systèmes de prévention d'intrusions "noyau" (KIDS/KIPS)

Les systèmes de détection d'intrusions « réseau » (NIDS)

Objectif : analyser de manière passive les flux en transit sur le réseau et détecter les intrusions en temps réel.

Un NIDS écoute donc tout le trafic réseau, puis l'analyse et génère des alertes si des paquets semblent dangereux.

Les systèmes de détection d'intrusions de type hôte (HIDS)

Un HIDS se base sur une unique machine, n'analysant cette fois plus le trafic réseau, mais l'activité se passant sur cette machine. Il analyse en temps réel les flux relatifs à une machine ainsi que les journaux.

Les systèmes de détection d'intrusions « hybrides »

Ils permettent de réunir les informations de diverses sondes placées sur le réseau. Leur appellation « hybride » provient du fait qu'ils sont capables de réunir aussi bien des informations provenant d'un système HIDS ou d'un système NIDS.

Il permettent de combiner plusieurs outils puissants tous ensemble pour permettre une visualisation centralisée des attaques.

Les systèmes de prévention d'intrusions (IPS)

Définition

ensemble de composants logiciels et matériels dont la fonction principale est d'empêcher toute activité suspecte détectée au sein d'un système.

Contrairement aux IDS simples, les IPS sont des outils aux fonctions « actives », qui en plus de détecter une intrusion, tentent de la bloquer.

Les systèmes de prévention d'intrusions « kernel » (KIDS/KIPS)

L'utilisation d'un détecteur d'intrusions au niveau noyau peut s'avérer parfois nécessaire pour sécuriser une station.

Exemple d'un serveur web : il serait dangereux qu'un accès en lecture/écriture dans d'autres répertoires que celui consultable via HTTP, soit autorisé. Grâce à un KIPS, tout accès suspect peut être bloqué directement par le noyau, empêchant ainsi toute modification pour le système.

Le KIPS peut également interdire l'OS d'exécuter un appel système qui ouvrirait un shell de commandes.

Puisqu'un KIPS analyse les appels systèmes, il ralentit l'exécution. C'est pourquoi ce sont des solutions rarement utilisées sur des serveurs souvent sollicités

2 approches principales pour les IDS :

- Par signature
- Par Comportement :
 - Détection d'anomalie
 - Vérification d'intégrité

Méthodes de détection par Signature

- Basé sur la reconnaissance de schémas déjà connus
- Utilisation d'expressions régulières
- Les signatures d'attaques connues sont stockées dans une base ;
et chaque événement est comparé au contenu de cette base
⇒ Si correspondance l'alerte est levée
- L'attaque doit être connue pour être détectée
- Peu de faux-positifs

Méthodes de détection par Signature

- Méthode la plus simple, basé sur :
Si EVENEMENT matche SIGNATURE Alors ALERTE



- Facile à implémenter pour tout type d'IDS L'efficacité des ids est liée à la gestion de leur base de signatures
 - MAJ
 - Nombre de règles
 - Signatures suffisamment précises
- Exemples :
 - Trouver le « motif /winnt/system32/cmd.exe » dans une requête http
 - Trouver le motif « failed su for root » dans un log système

Méthodes de détection par Signature

- **Avantage**

- Simplicité de mise en œuvre
- Rapidité de diagnostic
- Précision (en fonction des règles)
- Identification du procédé d'attaque :
 - Procédé
 - Cibles
 - Sources
 - Outils

- **Inconvénients :**

- Ne détecte que les attaques connues
- Maintenance de la base
- Techniques d'évasion possibles dès que les signatures sont connues

- Basée sur le comportement « normal » du système
- Une déviation par rapport à ce comportement est considérée suspecte
- Le comportement doit être modélisé : on définit alors un profil
- Une attaque peut être détectée sans être préalablement connue

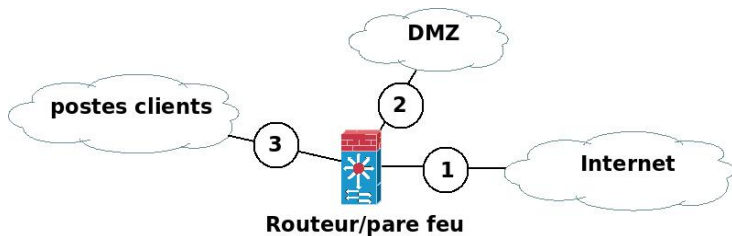
Vérification d'intégrité :

- Génération d'une somme de contrôle sur des fichiers d'un système
- Une comparaison est alors effectuée avec une somme de contrôle de référence
- Exemple : une page web
- Méthode couramment employée par les HIDS

Où placer un IDS/IPS

Dépend de ce que l'ont veut ?

- Voir les attaques (HoneyPot - pot de miel) : connaître les failles de sécurité
- surveiller les attaques sur un réseau :
 - Extérieur
 - Intérieur



- Position (1) : détection de toutes les attaques :
 - Log trop complet et analyse trop complexe :
 - bon pour un Honeypot
- Position (2) : détecte les attaques non filtrés par le pare feu
- Position (3) : Comme 2 + Attaque interne (80% des attaques sont de l'intérieur) :
 - Trojans
 - Virus
 - Etc.

Honeypot : un IDS particulier

Ordinateur ou programme volontairement vulnérable destiné à attirer et à piéger les pirates.

But :

- Occuper le pirate
- Découvrir de nouvelles attaques
- Garder le maximum de traces de l'attaque

Les 2 types de Honeypot

- ① **Faible interaction** : émulation de services sans réel système d'exploitation (exemple : Honeyd)
- ② **Forte interaction** : utilisation d'un réel système d'exploitation plus ou moins sécurisé

- Démon qui crée plusieurs hôtes virtuels sur le réseau
- Simule l'existence de services actifs sur les hôtes virtuels
- Toutes les connexions entrantes et sortantes sont enregistrées

AIDE (Advanced Intrusion Detection Environment)

AIDE (Advanced Intrusion Detection Environment) est un programme de détection d'intrusions. Il vérifie l'intégrité des fichiers.

Installation sous debian/ubuntu :

```
apt-get install aide
```

Configuration de «aide»

Fichier de configuration :

`/etc/aide/aide.conf`

Modifier les regles de bases (fonctionne comme des alias) :

```
# Custom rules
```

```
Binlib = p+i+n+u+g+s+b+m+c+md5
```

```
ConfFiles = p+i+n+u+g+s+b+m+c
```


Signification

p	permissions
i	inode
n	number of links
u	user
g	group
s	size
b	block count
m	mtime
a	atime
c	ctime
S	check for growing size
md5	md5 checksum

Format des lignes de conf

- Il y a une liste de répertoires à vérifier suivi des règles à appliquer
- '!' en début de ligne ignore le répertoire
- '=' en début de ligne ne descendra pas récursivement

Exemple

```
#conf  
/etc ConfFiles
```

Commandes de base et initialisation

- Pour créer la base de données initiale.

```
aide -i -c /etc/aide/aide.conf  
mv /var/lib/aide/aide.db.new /var/lib/aide/aide.  
db
```

- Pour contrôler et mettre à jour la base de données automatiquement

```
aide -u -c /etc/aide/aide.conf
```

C'est IDS réseaux (NIDS) :

- open source
- Conçu en 1998 par Marty Roesch racheté par SourceFire
- le plus très répandu des NIDS
- Il permet d'analyser les flux de données par rapport à une base de données de signatures (règles), mais aussi de détecter les anomalies.

Fonctionnalités :

- Permet d'interagir avec le pare-feu pour bloquer des intrusion (utilisation comme IPS) « snort natif, snort-inline, autres plugins »
- Possibilité de créer ses propres règles
- Multi-plateforme (existe sous Windows et Linux)
- Installation sous Debian/Ubuntu :
`apt-get install snort`

Snort est fourni avec une base de règles permettant de détecter un bon nombre d'attaques ou d'événements anormaux.

Il est possible d'écrire ses propres règles en suivant une syntaxe précise. Les règles de Snort peuvent « agir » sur le trafic (comme le fait **iptables**) en rejetant certains paquets.

Configuration de snort sous Linux

- Fichier de configuration :
/etc/snort/snort.conf
Configuration des variables pour le réseau
- Configuration des réseaux à écouter
- Configuration des services à surveiller (http, dns, ...)
- Configuration des plugins de sortie (MySQL, PostgreSQL, écran, ...)
- Choix des règles à utiliser :
Répertoire : **/etc/snort/rules/** (ensemble des signatures)

Exemple de règles :

- Pour détecter les tentatives de login sous l'utilisateur root, pour le protocole ftp (port 21) :

```
alert tcp any any -> 192.168.100.0/24 21 (  
  content: "USER root"; nocase; msg: "FTP root  
  user access attempt";)
```


Chapitre 5

Audit technique de sécurité

(source wikipedia)

Définition

L'audit informatique (en anglais Information Technology Audit ou IT Audit) a pour objectif d'identifier et d'évaluer les risques (opérationnels, financiers, de réputation notamment) associés aux activités informatiques d'une entreprise ou d'une administration.

Il existe deux grandes catégories d'audit :

- ❶ La première comporte les audits globaux d'entité durant lesquels toutes les activités ayant trait aux systèmes d'informations sont évaluées.
- ❷ La seconde catégorie correspond aux audits thématiques, ayant pour objectif la revue d'un thème informatique au sein d'une entité (la gestion de projet, la sécurité logique par exemple).

L'audit n'est pas à confondre avec l'activité de conseil qui vise, de manière générale, à améliorer le fonctionnement et la performance d'une organisation avec une éventuelle implication dans la mise en œuvre de cette amélioration.

La démarche d'audit informatique est générale et s'applique à différents domaines comme la fonction informatique, les études informatiques, les projets informatiques, l'exploitation, la planification de l'informatique, les réseaux et les télécommunications, la sécurité informatique, les achats informatiques, l'informatique locale ou l'informatique décentralisée, la qualité de service, l'externalisation, la gestion de parc, les applications opérationnelles ...

Le but de l'audit de la fonction informatique est de répondre aux préoccupations de la direction générale ou de la direction informatique concernant :

- l'organisation de la fonction informatique
- son pilotage
- ses relations avec les utilisateurs
- ses méthodes de travail
- ...

L'audit de l'exploitation a pour but de s'assurer que le ou les différents centres de production informatiques fonctionnent de manière efficace et qu'ils sont correctement gérés.

L'audit des projets informatiques est un audit dont le but est de s'assurer qu'il se déroule normalement et que l'enchaînement des opérations se fait de manière logique et efficace de façon qu'on ait de fortes chances d'arriver à la fin de la phase de développement à une application qui sera performante et opérationnelle.

L'audit d'un projet informatique ne se confond pas avec un audit des études informatiques.

Audit des applications opérationnelles

Les audits précédents sont des audits informatiques, alors que l'audit d'applications opérationnelles couvre un domaine plus large et s'intéresse au système d'information de l'entreprise. Ce sont des audits du système d'information. Ce peut être l'audit de l'application comptable, de la paie, de la facturation, . . . Mais, de plus en plus souvent, on s'intéresse à l'audit d'un processus global de l'entreprise comme les ventes, la production, les achats, la logistique, . . .

Il est conseillé d'auditer une application de gestion tous les deux ou trois ans de façon à s'assurer qu'elle fonctionne correctement et, le cas échéant pouvoir apporter les améliorations souhaitable à cette application ou à ce processus.

Définition

L'audit de sécurité d'un système d'information (SI) est une vue à un instant T de tout ou partie du SI, permettant de comparer l'état du SI à un référentiel.

L'audit répertorie les points forts, et surtout les points faibles (vulnérabilités) de tout ou partie du système. L'auditeur dresse également une série de recommandations pour supprimer les vulnérabilités découvertes. L'audit est généralement réalisé conjointement à une analyse de risques, et par rapport au référentiel.

Le référentiel est généralement constitué de :

- la politique de sécurité du système d'information (**PSSI**)
- la base documentaire du SI
- réglementations propre à l'entreprise
- textes de loi
- documents de référence dans le domaine de la sécurité informatique

Pourquoi un audit de sécurité ?

L'audit peut être effectué dans différents buts :

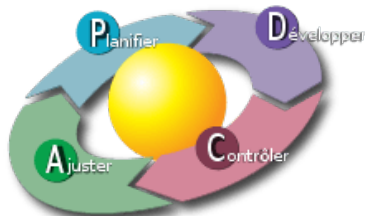
- réagir à une attaque
- se faire une bonne idée du niveau de sécurité du **SI**
- tester la mise en place effective de la **PSSI**
- tester un nouvel équipement
- évaluer l'évolution de la sécurité (implique un audit périodique)

But

- Il a pour but de vérifier la sécurité.
- Dans le cycle de sécurisation, la vérification intervient après la réalisation d'une action.
- Par exemple, lors de la mise en place d'un nouveau composant dans le SI, il est bon de tester sa sécurité après avoir intégré le composant dans un environnement de test, et avant sa mise en œuvre effective.

⇒ La **roue de Deming** illustre ce principe.

Roue de Deming (de l'anglais Deming wheel)



La roue de Deming est un moyen mnémotechnique qui permet de repérer avec simplicité les étapes à suivre pour améliorer la qualité dans une organisation.

Roue de Deming : démarche d'utilisation

La méthode comporte quatre étapes, chacune entraînant l'autre, et vise à établir un cercle vertueux. Sa mise en place doit permettre d'améliorer sans cesse la qualité d'un produit, d'une œuvre, d'un service, etc.

- ❶ **Plan** : Préparer, planifier (ce que l'on va réaliser) ;
- ❷ **Do** : Développer, réaliser, mettre en œuvre (le plus souvent, on commence par une phase de test) ;
- ❸ **Check** : Contrôler, vérifier ;
- ❹ **Act** (ou Adjust) : Agir, ajuster, réagir (si on a testé à l'étape Do, on déploie lors de la phase Act).

Pour arriver à dresser une liste la plus exhaustive possible des vulnérabilités d'un système, différentes pratiques existent et sont traditionnellement mises en œuvre :

- Interviews
- Les relevés de configuration
- L'audit de code
- Fuzzing
- **Les tests d'intrusion**

Ils sont généralement essentiels à tout audit. Dans le cas où l'organisation du SI est analysée, ils sont même indispensables. Toutes les personnes ayant un rôle à jouer dans la sécurité du SI sont à interroger :

- Le directeur des systèmes d'information (DSI)
- Le ou les responsable(s) de la sécurité des systèmes d'information (RSSI)
- Les administrateurs
- Les utilisateurs du système d'information
- Tout autre rôle ayant un lien avec la sécurité

Il est important de bien formuler les questions pour que les personnes ne se sentent pas jugées.

Les relevés de configuration

- Il s'agit ici d'analyser, profondément, les composants du système d'information.
- Les configurations sont inspectées dans les moindres détails.

⇒ Suite à cette observation, la liste des vulnérabilités est dégagée en comparant le relevé à des configurations réputées sécurisée et à des ensembles de failles connues.

Les relevés de configuration

Tout peut être inspecté, allant de l'architecture du SI aux applications, en passant par les hôtes (clients et serveurs). Par exemple sur un serveur, on va analyser :

- le chargeur de démarrage,
- les mécanismes d'authentification (robustesse des mots de passe, utilisation d'authentification forte...),
- le système de fichiers (droits d'accès, utilisation de chiffrement...),
- les services
- la journalisation,
- la configuration réseau,
- ...

- Il existe des bases de vulnérabilités très fiables pour les applications répandues.
- Pour des applications moins utilisées, ou codées par l'entreprise elle-même, il peut être nécessaire d'analyser leur sécurité. Si les sources de l'application sont disponibles, il faut lire et comprendre le code source, pour déceler les problèmes qui peuvent exister. Notamment, les dépassements de tampon (buffer overflow), les bugs de format, ou pour une application web, les vulnérabilités menant à des injections SQL ...

L'audit de code est une pratique très fastidieuse et longue !

- Pour les applications boîte noire, où le code n'est pas disponible, il existe une alternative à l'analyse de code, qui est le **fuzzing**.
- Cette technique consiste à analyser le comportement d'une application en injectant en entrée des données plus ou moins aléatoires, avec des valeurs limites.
- Contrairement à l'audit de code qui est une analyse structurelle, le fuzzing est une analyse **comportementale** d'une application.

Définition

Un test d'intrusion (« penetration test » ou « pentest » en anglais) est une méthode d'évaluation de la sécurité d'un système ou d'un réseau informatique.

Les tests d'intrusion : méthode

La méthode consiste généralement à simuler une attaque d'un utilisateur mal intentionné, voire d'un logiciel malveillant. On analyse alors :

- les risques potentiels dus à une mauvaise configuration d'un système
- d'un défaut de programmation
- d'une vulnérabilité liée à la solution testée.

Le principal but de cette manœuvre est de trouver des vulnérabilités exploitables en vue de proposer un plan d'actions permettant d'améliorer la sécurité d'un système.

Test d'intrusion vs Audit de sécurité

- La différence avec un simple audit de sécurité est la motivation pour la personne à aller jusqu'à exploiter les failles, montrant ainsi la vulnérabilité.
- L'exploitation n'a bien sûr pas pour but de détruire ou endommager le système, mais elle permettra de situer le degré du risque lui étant associé.

L'analyse peut se réaliser selon trois cas :

- ❶ Le testeur se met dans la peau d'un attaquant potentiel, et ne possède aucune information ;
- ❷ Le testeur possède un nombre limité d'informations (exemple : un compte) ;
- ❸ Le testeur possède les informations dont il a besoin.

Le testeur n'a aucune information (ou black box)

Fondement :

- Il s'agit dans un premier temps de rechercher des informations sur l'entreprise, la personne, ou toute autre donnée pour s'assurer que la cible est bien celle que l'on tente d'infiltrer.
- Connaître la situation géographique, les informations générales d'une société, ou son fournisseur d'accès à Internet.

Le testeur n'a aucune information (ou black box)

Pour cela le testeur dispose de plusieurs outils :

- Le web : pour connaître l'adresses email, les adresses postales, numéros de téléphone, ... sur une cible donnée (entité physique ou morale).
- Le service DNS via les outils **nslookup** et **dig** afin d'interroger les serveurs DNS pour obtenir soit l'adresse IP en fonction d'un nom de domaine, soit l'inverse.
- Le service **whois** qui permet de récupérer des informations diverses sur une adresse IP ou un nom de domaine.

Ensuite pouvoir schématiser l'emplacement et l'étendue du système à tester, c'est-à-dire réaliser une cartographie (ou map en anglais).

Le fait d'effectuer une telle analyse permet de comprendre le mode de fonctionnement et le raisonnement de son propriétaire. De plus, un système en réseau étendu nécessite une sécurité plus importante :

⇒ la pénétration d'un seul ordinateur d'un réseau peut permettre la pénétration de tous les autres beaucoup plus facilement.

La compromission (exploitation d'une vulnérabilité, **escalade de privilège** et mise en place d'un **rootkit**) d'un seul poste permet :

- La mise en place d'un sniffer qui permet quant à elle la récupération des identifiants et mots de passe pour les protocoles de communication en clair.
- Une cartographie du réseau de l'intérieur et un inventaire des ressources disponibles de ce fait beaucoup plus simple et détaillé en utilisant par exemple les propriétés des protocoles SNMP, RPC et SMB.
- La mise en place d'une détection automatisée de vulnérabilités ...
- L'effacement des traces.

Élévation des privilèges

Définition

Une élévation des privilèges est un mécanisme permettant à un utilisateur d'obtenir des privilèges supérieurs à ceux qu'il a normalement.

- Généralement, un utilisateur va vouloir élever ses privilèges pour devenir administrateur du système, afin d'effectuer des tâches qu'il n'a pas le droit de faire en temps normal.
- Ce mécanisme est utile pour lancer des processus sensibles, pouvant nécessiter des compétences particulières en administration système : par exemple lors d'une manipulation des partitions d'un disque dur, ou lors du lancement d'un nouveau service.

Définition (www.dicodunet.com)

Le rootkit est un ensemble de programmes destinés à compromettre un système afin d'en obtenir et d'y maintenir un accès root (administrateur).

- Ils sont dans la majorité des cas capables d'infiltrer le noyau (kernel) du système d'exploitation afin de se dissimuler et d'intercepter les appels système afin de renvoyer des réponses truquées aux autres logiciels.
- Étant cachés dans les couches basses du système d'exploitations, les rootkits sont indétectable par les outils de sécurité standards, comme les antivirus.
- Ils sont très difficiles à repérer, seulement en effectuant un contrôle attentif du fonctionnement des processus et des connexions.

Le testeur n'a aucune information : cartographie du réseau

Il s'agit principalement de :

- Prise d'empreinte de la pile TCP/IP afin d'étudier les différentes réponses dues aux implémentations des piles TCP/IP et de déterminer le système d'exploitation installé, ainsi que sa version.
- Balayage des ports afin de détecter des ports ouverts et les règles de filtrage des machines.
- Identifier les services qui tournent derrière ces ports et leur versions en vue d'une exploitation ultérieure.

L'outil **Nmap** permet de réaliser ces opérations.

Ces techniques peuvent être aisément détectées à l'aide d'un système de détection d'intrusion (IDS).

Le testeur n'a aucune information : Applicatif

Après avoir trouvé les programmes actifs qui communiquent avec un autre réseau, trouver une faille dans ces applications peut amener à corrompre tout un système en peu de temps.

Le but recherché ici va être de corrompre une application pour lui faire exécuter son propre code, (généralement donné en langage d'assemblage). La plus grande faille connue à ce jour est le dépassement de tampon.

Définition

Un dépassement de tampon ou débordement de tampon (en anglais, buffer overflow) est un bug par lequel un processus, lors de l'écriture dans un tampon, écrit à l'extérieur de l'espace alloué au tampon, écrasant ainsi des informations nécessaires au processus.

Cette technique est couramment utilisée par les pirates. La stratégie de l'attaquant est alors de détourner le programme bogueé en lui faisant exécuter des instructions qu'il a introduites dans le processus.

Le testeur possède un nombre limité d'informations (ou grey box)

En général, lors de tests d'intrusion en mode boîte grise, le testeur dispose uniquement d'un couple (identifiant - mot de passe). Ceci lui permet notamment de passer l'étape d'authentification.

L'objectif de ce type de test est d'évaluer le niveau de sécurité vis-à-vis d'un "utilisateur normal".

Le testeur se trouve en possession des informations nécessaires (ou white box)

Le testeur peut être en possession de nombreuses informations. Parmi elles, les plus courantes sont :

- Schémas d'architecture ;
- Compte utilisateur permettant de s'authentifier ;
- Code source de l'application ;
- ...

Dans ce cas, il n'aura plus qu'une chose à faire : rechercher la ou les failles, et trouver le moyen de les exploiter.

De même, un testeur se trouvant à l'intérieur du réseau à tester aura plus de facilité à trouver ces failles car il connaît non seulement le système, mais il peut avoir accès directement aux ressources dont il a besoin.

Les tests Red Team

- Ils ont pour objectif de simuler le scénario où un pirate souhaiterait pénétrer le système d'information d'une entreprise ou d'une institution sans limite de temps, ni de périmètre.
- Ils se déroulent sur une période de temps plus longue qu'un test d'intrusion normal (2 à 3 mois, contre 1 à 2 semaines) et n'ont pas de périmètre précis défini par le commanditaire (le testeur démarre avec uniquement le nom de l'entreprise).

Outils de tests de pénétration (Pen Test)

C'est un scanner de ports. Il est conçu pour détecter les ports ouverts, identifier les services hébergés et obtenir des informations sur le système d'exploitation d'un ordinateur distant. Ce logiciel est devenu une référence pour les administrateurs réseaux car l'audit des résultats de Nmap fournit des indications sur la sécurité d'un réseau. Il est disponible sous Windows, Mac OS X, Linux, BSD et Solaris.

zenmap est une version graphique de **nmap** (lien :

<https://nmap.org/zenmap/>).

Nikto (Linux)/Wikto(Windows) est un scanner de serveurs web.

sqlmap (<http://sqlmap.org/>)

Sqlmap est un outil de sql injection. On lui fournit une url avec un paramètre, il va se charger de tester les injections SQL que l'on peut faire dessus.

Metasploit est un projet (open-source) en relation avec la sécurité des systèmes informatiques. Son but est de fournir des informations sur les vulnérabilités de systèmes informatiques, d'aider à la pénétration et au développement de signatures pour les IDS.

Lien téléchargement : <https://www.rapid7.com/products/metasploit/download/community/>