

# SÉCURITÉ DES RÉSEAUX ET DES ARCHITECTURES IT

## Intrusion Detection System (IDS)

Prof. R.Lamrani Alaoui

[lamrani.alaaoui.rokia@ucd.ac.ma](mailto:lamrani.alaaoui.rokia@ucd.ac.ma)

1

## PLAN DU COURS

- Introduction
- Types de détection d'intrusions
- Précision des systèmes de détection d'intrusions
- Techniques de détection d'intrusions
- Autres techniques de détection d'intrusions
- Exemple d'un IDS/IPS: Snort

2

## INTRODUCTION

- Il existe des méthodes pour éviter certaines attaques dans un réseau local:
  - ARP cache poisoning / IP Spoofing => authentification
  - MiTM/ Sniffing => TLS, SSH, SFTP, VPN, ...
  - SYN Flood => SYN cookie
  - DHCP Spoofing => DHCP Snooping
  - DNS Spoofing => DNSSEC
  - Evil Twin => WPA-Enterprise
  - Injection SQL => Requêtes préparées
  - Firewall
  - Etc
- Mais il est toujours possible que les attaques réussissent à contourner ces défenses (il n'y a pas de sécurité absolue)
- Donc, il est recommandé de détecter les attaques exécutées ou en cours d'exécution (detect if you can't prevent).
- Pour cela, on utilise des systèmes de détection d'intrusion (IDS) pour pouvoir détecter une attaque passée, ou des systèmes de prévention d'intrusion (IPS) pour stopper des attaques en cours d'exécution.

3

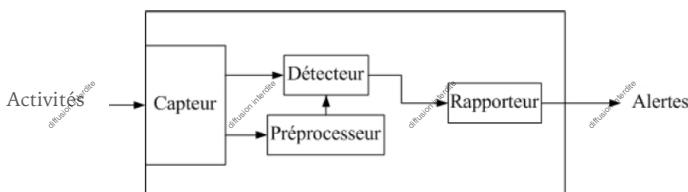
## INTRODUCTION

- Une intrusion est un incident de sécurité où un attaquant externe (outsider) tente ou réussit à obtenir un accès à un système ou un réseau sans autorisation.
- Une extrusion est un incident de sécurité où un attaquant interne (insider) tente ou réussit à obtenir un accès à un système ou un réseau sans autorisation.
- Un système de détection d'intrusions (IDS) est un système (software+hardware) qui automatise le processus d'analyse des activités qui se produisent au niveau d'un ordinateur ou d'un réseau pour y détecter des signes de problèmes de sécurité et les signaler à l'administrateur de sécurité.
- Les IDS peuvent détecter aussi des extrusions, mais la plupart des IDS sont conçus pour détecter les intrusions.

4

## INTRODUCTION

- Composants et fonctionnement d'un IDS:

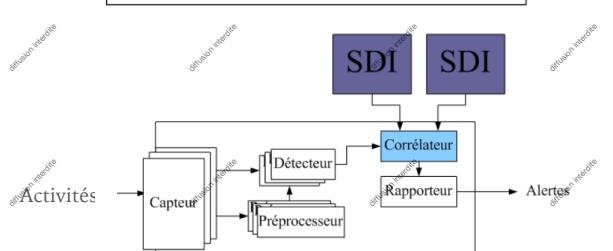


5

## INTRODUCTION

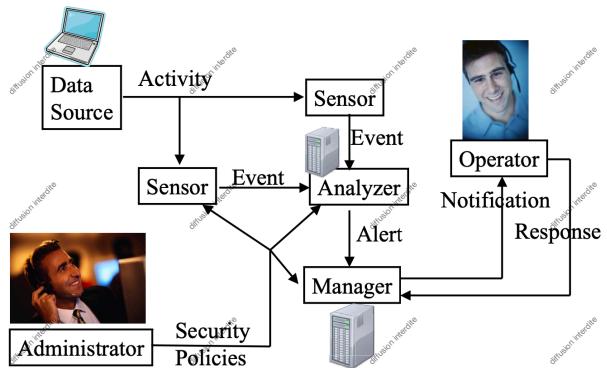
- Composants et fonctionnement d'un IDS:
  1. Collection des données: paquets réseau, logs systèmes,..
  2. Pré-Processing: supprimer les informations inutiles, formater..
  3. Détection: appliquer des méthodes de détection d'intrusion sur les données
  4. Génération des alertes: envoyer des alertes si une activité suspecte ou malicieuse a été détectée
  5. Réponses: Réagir aux alertes envoyées (ex. Notifier les administrateurs, Ecrire dans les logs, Bloquer une IP, Isoler un hôte compromis..)

6



## INTRODUCTION

- Security Event & Response Pipeline (terminologie des années 2000):



7

## INTRODUCTION

- Security Event & Response Pipeline (modern terminology):
1. Capteurs --> IDS, IPS, Firewall, Hôtes... génèrent des alertes et/ou des logs suite à la détection d'une intrusion
  2. Analyseur --> Le SIEM (Security Information and Event Management) fait la corrélation entre les différents événements générés par les capteurs, et si une intrusion est confirmée, il émet des alertes qu'il est possible de consulter via la console de SIEM (Manager)
  3. Opérateur --> Le SOC (Security Operations Center) est informé par ces alertes via la console SIEM (ou par email..), l'analyste SOC analyse ensuite les alertes et décide si une réponse (organisationnelle ou technique) est nécessaire.



8

## INTRODUCTION

- Security Event & Response Pipeline (modern terminology):
1. Certains capteurs envoient les événements au SIEM en utilisant un protocole et un format de message bien défini (ex.Syslog)
  2. Autre capteurs peuvent avoir une API REST que le SIEM peut interroger pour obtenir les événements au format JSON.
  3. Le SOC émet les réponses via SOAR (Security Orchestration, Automation and Response) qui les traduit dans le format attendu par le capteur destinataire.
  4. Et avant tout, l'administrateur définit la politique de sécurité au niveau du SIEM, et au niveau des capteurs



9

## TYPES DE DÉTECTION D'INTRUSIONS

- Il existe 2 types principaux de détection d'intrusions:

  - NIDS (Network Intrusion Detection Systems)
  - HIDS (Host Intrusion Detection Systems)

- Il existe d'autres types de détection d'intrusions:

  - PIDS (Protocol Intrusion Detection Systems)
  - APIDS (Application Protocol Intrusion Detection Systems)
  - Hybrid Intrusion Detection Systems

- La différence principale entre ces types de détection est où l'IDS est déployés.



10

## TYPES DE DÉTECTION D'INTRUSIONS: NIDS

- Le NIDS analyse le trafic réseau paquet par paquet en temps réel ou périodiquement (ex. Snort, Suricata).
- Le NIDS peut être installé:
  - **in-line:** inséré dans un segment de réseau de telle façon que tout le trafic réseau passe par le NIDS
  - **out-of-band:** surveille une copie du trafic réseau (SPAN Port).
- Le NIDS est généralement placé derrière le Firewall à la frontière du réseau qu'on cherche à protéger
- Le NIDS alerte lorsqu'il détecte une intrusion, mais ne bloque pas l'intrusion qu'il soit installé in-line ou out-of-band. S'il est placé in-line et qu'il bloque les intrusions il est alors appelé IPS (Intrusion Prevention System)



9

## TYPES DE DÉTECTION D'INTRUSIONS: NIDS

- NIDS out-of-band:

```

graph LR
    DS[Data Source] -- "diffusion interne" --> S1[Sensor]
    DS -- "diffusion interne" --> S2[Sensor]
    DS -- "diffusion interne" --> AP[Administrator]
    S1 -- "diffusion interne" --> S2
    S1 -- "diffusion interne" --> A[Analyzer]
    S2 -- "diffusion interne" --> A
    A -- "diffusion interne" --> M[Manager]
    M -- "diffusion interne" --> N[Notification]
    M -- "diffusion interne" --> R[Response]
    O -- "diffusion interne" --> R
    AP -- "diffusion interne" --> SP[Security Policies]
    SP -- "diffusion interne" --> S2
    
```

This diagram shows the deployment of NIDS out-of-band. It features a Data Source (laptop icon) sending events to two Sensors (monitor icons). One Sensor also receives events from an Administrator (person icon). Both Sensors send events to an Analyzer (server icon). The Analyzer sends alerts to a Manager (server icon), which then triggers Notifications (person icon) and Responses. The Manager also receives notifications from the Operator (person icon). The Administrator defines Security Policies (document icon) which influence the second Sensor.

11

## TYPES DE DÉTECTION D'INTRUSIONS: NIDS

- NIDS in-line:

```

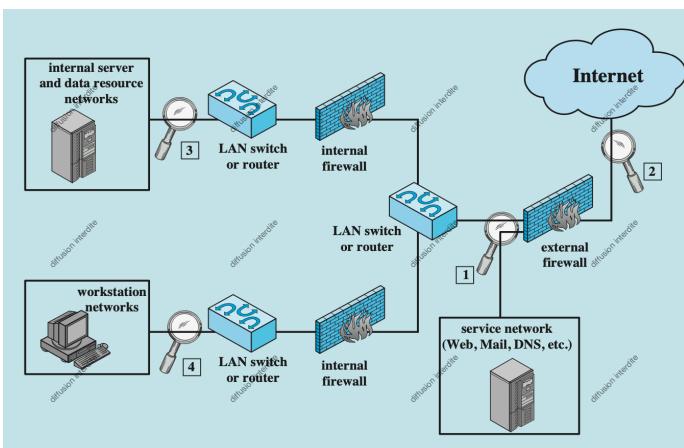
graph LR
    DS[Data Source] -- "diffusion interne" --> S1[Sensor]
    DS -- "diffusion interne" --> S2[Sensor]
    DS -- "diffusion interne" --> AP[Administrator]
    S1 -- "diffusion interne" --> S2
    S1 -- "diffusion interne" --> A[Analyzer]
    S2 -- "diffusion interne" --> A
    A -- "diffusion interne" --> M[Manager]
    M -- "diffusion interne" --> N[Notification]
    M -- "diffusion interne" --> R[Response]
    O -- "diffusion interne" --> R
    AP -- "diffusion interne" --> SP[Security Policies]
    SP -- "diffusion interne" --> S2
    
```

This diagram shows the deployment of NIDS in-line. It features a Data Source (laptop icon) sending events to two Sensors (monitor icons). One Sensor also receives events from an Administrator (person icon). Both Sensors send events to an Analyzer (server icon). The Analyzer sends alerts to a Manager (server icon), which then triggers Notifications (person icon) and Responses. The Manager also receives notifications from the Operator (person icon). The Administrator defines Security Policies (document icon) which influence the second Sensor.

12

## TYPES DE DÉTECTION D'INTRUSIONS: NIDS

- Emplacements possibles d'un NIDS:



## TYPES DE DÉTECTION D'INTRUSIONS: NIDS

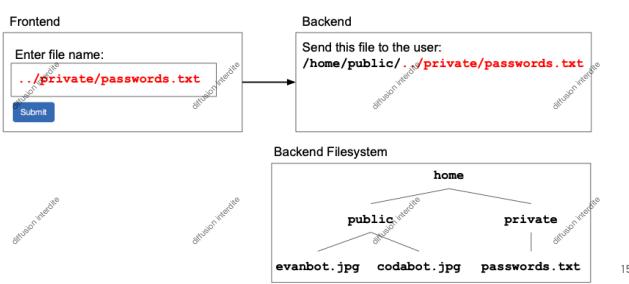
- Avantages:

- Pas coûteux: un seul NIDS peut protéger plusieurs ordinateurs ou machines.
- Scalable: si d'autres systèmes sont ajoutés, il suffit de renforcer la capacité de calcul du NIDS (mémoire, CPU,..)
- Facile à administrer: il est facile d'installer et administrer un seul NIDS
- Il n'affecte pas les systèmes déjà existants dans le réseau, et il ne consomme pas de ressources sur les postes client ou serveurs.
- On a pas besoin d'avoir confiance à plusieurs systèmes: Faire confiance au NIDS seulement.

14

## TYPES DE DÉTECTION D'INTRUSIONS: NIDS

- Inconvénients:
- Interprétation ambiguë: un attaquant peut exploiter la différence d'interprétation entre le NIDS et le destinataire pour s'évader du NIDS
- Exemple (Attaque Path Traversal):

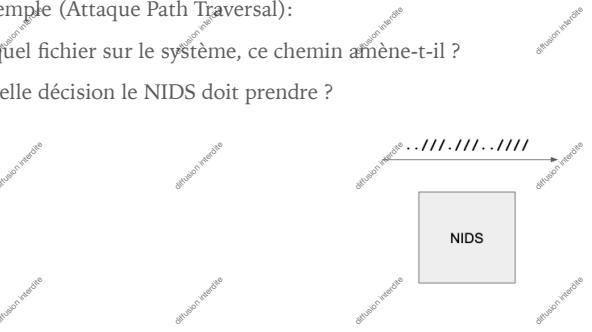


15

## TYPES DE DÉTECTION D'INTRUSIONS: NIDS

- Inconvénients:

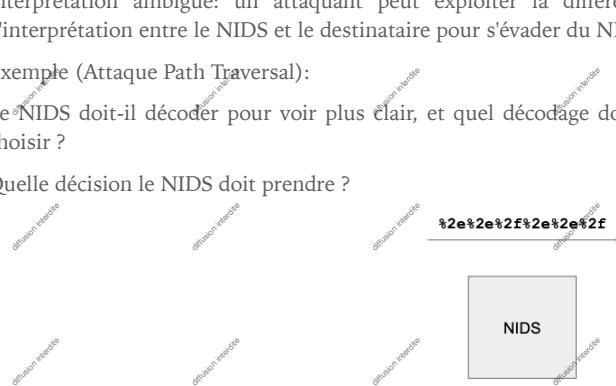
- Interprétation ambiguë: un attaquant peut exploiter la différence d'interprétation entre le NIDS et le destinataire pour s'évader du NIDS
- Exemple (Attaque Path Traversal):
- A quel fichier sur le système, ce chemin amène-t-il ?
- Quelle décision le NIDS doit prendre ?



16

## TYPES DE DÉTECTION D'INTRUSIONS: NIDS

- Inconvénients:
- Interprétation ambiguë: un attaquant peut exploiter la différence d'interprétation entre le NIDS et le destinataire pour s'évader du NIDS
- Exemple (Attaque Path Traversal):
- Le NIDS doit-il décoder pour voir plus clair, et quel décodage doit-il choisir ?
- Quelle décision le NIDS doit prendre ?



## TYPES DE DÉTECTION D'INTRUSIONS: NIDS

- Inconvénients:

- Interprétation ambiguë: un attaquant peut exploiter la différence d'interprétation entre le NIDS et le destinataire pour s'évader du NIDS
- Exemple (Attaque Path Traversal):
- L'attaque ressemble à une attaque Path Traversal, le NIDS doit alerter, mais que se passe-t-il si le paquet n'est pas fait pour arriver au destinataire ?
- Quelle décision le NIDS doit prendre ?



17

## TYPES DE DÉTECTION D'INTRUSIONS: NIDS

- Inconvénients:
- Interprétation ambiguë: un attaquant peut exploiter la différence d'interprétation entre le NIDS et le destinataire pour s'évader du NIDS
- Exemple (Attaque de Fragmentation):
- Le NIDS reçoit des fragments, aucun fragment ne correspond à une signature d'attaque. Mais, une fois les fragments sont reconstitués par le destinataire, ils forment un paquet malicieux.
- Quelle décision le NIDS doit prendre ?



19

## TYPES DE DÉTECTION D'INTRUSIONS: NIDS

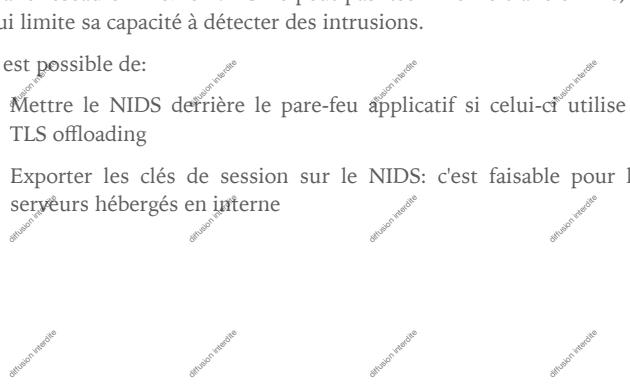
- Inconvénients:
- Interprétation ambiguë: un attaquant peut exploiter la différence d'interprétation entre le NIDS et le destinataire pour s'évader du NIDS
- Pour éviter les attaques d'évasion, il est possible de:
  - Faire en sorte que le NIDS et le destinataire de paquet interprètent les données de la même façon.
  - Considérer toutes les interprétations possibles au lieu d'une seule.
  - Envoyer des alertes pour toute tentative d'évasion potentielle



20

## TYPES DE DÉTECTION D'INTRUSIONS: NIDS

- Inconvénients:
- Trafic réseau chiffré: le NIDS ne peut pas déchiffrer le trafic chiffré, ce qui limite sa capacité à détecter des intrusions.
- Il est possible de:
  - Mettre le NIDS derrière le pare-feu applicatif si celui-ci utilise le TLS offloading
  - Exporter les clés de session sur le NIDS: c'est faisable pour les serveurs hébergés en interne



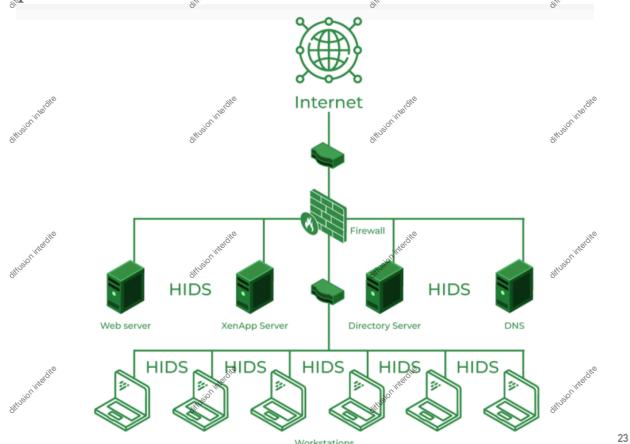
21

## TYPES DE DÉTECTION D'INTRUSIONS: HIDS

- Le HIDS s'installe sur chaque machine ou ordinateur de réseau (ex. OSSEC).
- Le HIDS analyse les informations provenant de l'OS:
  - Les logs de systèmes et des applications (ex. Application web)
  - Les activités du système (login/logout, accès aux fichiers, utilisation de ressources..)
  - La base de registre, l'espace disque..
  - Les interactions entre applications
  - Possiblement l'exécution des applications en mémoire
  - ...

## TYPES DE DÉTECTION D'INTRUSIONS: HIDS

- Emplacement d'un HIDS:



23

## TYPES DE DÉTECTION D'INTRUSIONS: HIDS

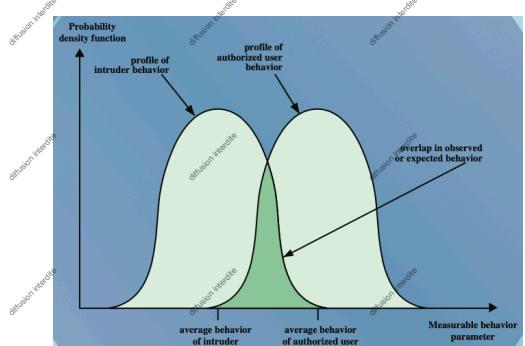
- Avantages:
  - Il est plus informé sur les attaques possibles puisqu'il est spécifique à un OS
  - En temps normal, en comparaison avec le NIDS, il a moins de données à analyser puisqu'il protège un seul système
  - Pas besoin d'ajouter un nouveau hardware sur le réseau.
  - Il n'est pas vulnérable aux attaques d'évasion.
- Inconvénients:
  - Couteux: Il faut un HIDS sur chaque machine de réseau
  - Consomme les ressources de la machine
  - Difficile à déployer
  - Il peut être désactivé ou attaqué si le poste client ou serveur est compromis.
  - Il peut être vulnérable aux attaques d'injection de code



24

## PRÉCISION DES SYSTÈMES DE DÉTECTION D'INTRUSIONS

- On suppose que les activités réalisées par un attaquant sont différentes des activités réalisées par des utilisateurs légitimes.
- Mais, s'il y a un chevauchement entre les activités observées et les activités attendues, il peut y avoir des erreurs de détection:



25

## PRÉCISION DES SYSTÈMES DE DÉTECTION D'INTRUSIONS

- Il existe deux types d'erreurs de détection:
  - Faux Positifs: l'IDS génère une alerte alors qu'il n'y a pas d'intrusions.
  - Faux Négatifs: l'IDS ne génère pas d'alertes alors qu'il y a une intrusion.
- La précision des IDS est souvent mesurée par le taux d'apparition de ces erreurs:
  - Taux de faux positifs: la probabilité qu'un IDS alarme sur une fausse intrusion
  - Taux de faux négatifs: la probabilité qu'un IDS n'alarme pas sur une vraie intrusion.



26

## PRÉCISION DES SYSTÈMES DE DÉTECTION D'INTRUSIONS

- IDS parfait: un casse-tête
- Est-il possible de construire un IDS qui a un FPR = 0% ?

```
void detector_with_no_false_positives(char *input) {
    printf("Nope, not an attack!");
}
```

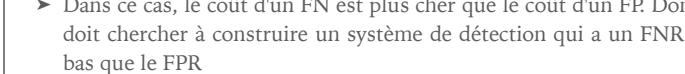
- Est-il possible de construire un IDS avec FNR = 0% ?

```
void detector_with_no_false_negatives(char *input) {
    printf("Yep, it's an attack!");
}
```

27

## PRÉCISION DES SYSTÈMES DE DÉTECTION D'INTRUSIONS

- IDS parfait => un casse-tête
- Pour améliorer la précision de détection d'un IDS, il faut chercher à diminuer le FPR et le FNR, avec une préférence de l'un ou l'autre selon la nature du système qu'on cherche à protéger:
  - Exemple (Détecteur d'incendies):
    - Coût d'un FP: Il faut vérifier tout le bâtiment pour trouver la source de l'incendie.
    - Coût d'un FN: Tout le bâtiment se brûle
  - Dans ce cas, le coût d'un FN est plus cher que le coût d'un FP. Donc on doit chercher à construire un système de détection qui a un FNR plus bas que le FPR



28

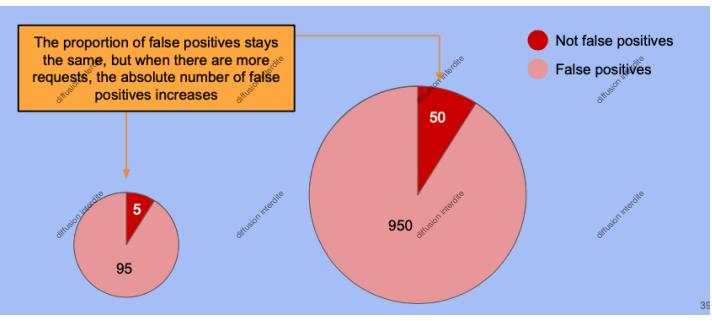
## PRÉCISION DES SYSTÈMES DE DÉTECTION D'INTRUSIONS

- IDS parfait => Le taux de base des attaques
- Exemple: Considérons un IDS avec un FPR = 0.1%
- Scénario 1: Un serveur reçoit 1000 requêtes normales et 5 requêtes malicieuses par jour
  - FPR attendu par jour est  $1000 * 0.1 = 1$  (une fausse alarme)
- Scénario 2: Un serveur reçoit 10.000.000 requêtes normales et 5 requêtes malicieuses par jour
  - FPR attendu par jour est  $10.000.000 * 0.1 = 10000$
  - Ca pourrait coûter cher si la vérification des FP coûte de l'argent
- L'IDS n'a pas changé, juste le taux de base des attaques qui a changé ! => Il devient difficile de détecter des intrusions si le taux de base des attaques est faible

29

## PRÉCISION DES SYSTÈMES DE DÉTECTION D'INTRUSIONS

- IDS parfait => Le taux de base des attaques



30

## PRÉCISION DES SYSTÈMES DE DÉTECTION D'INTRUSIONS

- IDS parfait => Erreur du taux de base (Base-Rate Fallacy)
- Exemple: Considérons l'IDS avec FPR = 0.1% et FNR=0%
- Si le serveur reçoit 10.000.000 requêtes normales et 5 requêtes malicieuses par jour, alors le FPR = 10.000 par jour
- Donc, sur un totale de 10.005 alarmes, seulement 5 correspondent à une réelle intrusion.
- Base-Rate Fallacy: Même si l'IDS envoie des alertes, il est peu probable de trouver l'intrusion, car le FPR est élevé

31

## PRÉCISION DES SYSTÈMES DE DÉTECTION D'INTRUSIONS

- Pour s'approcher à un IDS parfait, il est possible de combiner les IDS:
- Envoyer une alerte si l'un des IDS détecte une intrusion:
  - FNR bas mais FPR élevé !
- Envoyer une alerte si tous les IDS détectent une intrusion:
  - FNR élevé mais FPR bas !
- Envoyer une alerte suivant une relation de corrélation entre les différents IDS, qui peut réaliser une bonne balance entre le FPR et le FNR.

32

## TECHNIQUES DE DÉTECTION D'INTRUSIONS

- Les techniques de détection d'intrusions décrivent comment l'IDS analyse les données pour trouver une intrusion.
- Il existe 4 techniques de détection d'intrusion:
  - Détection par signature (misuse detection, ou blacklist)
  - Détection par spécification (whitelist)
  - Détection d'anomalies (anomaly detection)
  - Détection par comportement (behavioral detection)

33

## TECHNIQUES DE DÉTECTION D'INTRUSIONS: PAR SIGNATURE

- La détection basée sur la signature utilise une liste de patterns (signatures) non autorisés (blacklist ou denylist), et alerte si l'un de ces patterns est retrouvé dans les données observées.
- La signature peut porter sur:
  - L'en-tête de la couche réseau ou transport (IP, ICMP, TCP..)
  - Les URLs
  - Le corps de la requête HTTP (DPI)
  - Les logs
- Différents langages existent pour décrire les signatures d'attaques, ainsi que des algorithmes pour la recherche et la comparaison de patterns.

34

## TECHNIQUES DE DÉTECTION D'INTRUSIONS: PAR SIGNATURE

- Exemples:
- Signature: la chaîne `./` est souvent utilisée dans une attaque Path Traversal
  - Donc, l'IDS alerte si il trouve cette chaîne dans le corps d'une requête HTTP.
- Signature: une connexion semi-ouverte est souvent utilisée dans une attaque DoS sur un serveur.
  - Donc, l'IDS garde un état des connexions, et si une connexion reste semi-ouverte pour un certain délai, il envoie une alerte.
- Signature: plusieurs Event ID = **4771** pour le même compte dans les Windows Event Logs, est souvent signe d'une attaque de brute force.
  - Donc, l'IDS alerte si l'événement avec l'ID **4771** se répète plusieurs fois pour le même compte.

35

## TECHNIQUES DE DÉTECTION D'INTRUSIONS: PAR SIGNATURE

- Avantages:
  - Simple et efficace pour la détection d'attaques connues
  - Utile pour le partage de bases de signatures d'attaques.
  - Taux faible de FP
- Inconvénients:
  - Ne détecte pas les attaques sans signatures connues (zero-day attacks)
  - Ne détecte pas les variantes des attaques connues si la variante ne se trouve dans la base de signatures
  - Il faut mettre à jour régulièrement la base de signatures

36

## TECHNIQUES DE DÉTECTION D'INTRUSIONS: PAR SPECIFICATION

- La détection basée sur la spécification utilise une liste des patterns autorisés (whitelist ou allowlist), et alerte si elle observe un pattern qui n'est pas dans cette liste.
- Pareil que la détection par signature, la détection basée sur la spécification peut porter sur les différents champs d'un paquet réseau ou sur d'autres données collectées par l'IDS. Aussi, des langages spécifiques sont utilisés pour décrire les spécifications et des algorithmes sont utilisés pour la recherche et la comparaison des patterns.
- Exemple:
- Spécification: le champ "Nom Fichier" doit contenir seulement des caractères alphanumériques (a-z,A-Z,0-9)
- Donc, l'IDS alerte si une requête contient des caractères non alphanumériques dans le champ "Nom Fichier".

37

## TECHNIQUES DE DÉTECTION D'INTRUSIONS: PAR SPECIFICATION

- Avantages:
  - Il permet de détecter des attaques zero-day
  - Il permet de réaliser un taux faible de FP si on spécifie en détail les patterns autorisés. Sinon, on risque d'avoir le résultat inverse.
- Inconvénients:
  - Peut prendre beaucoup de temps et d'efforts pour spécifier manuellement tout ce qui est autorisé.
  - Faut mettre à jour les spécifications pour chaque changement au niveau du système surveillé.

38

## TECHNIQUES DE DÉTECTION D'INTRUSIONS: PAR ANOMALIE

- La détection d'anomalies consiste à développer un modèle des données normales, et à alerter sur chaque donnée qui dévie de la normale.
- La détection d'anomalies ressemble à la détection basée sur les spécifications, sauf que dans la deuxième on spécifie manuellement ce qui est normal, alors que dans la première on utilise le ML/DL pour développer un modèle qui peut identifier ce qui est normal.
- Deux phases:
  - Phase d'apprentissage: apprendre le profil normal des données
  - Phase de détection: le modèle entraîné classifie les données comme normales ou anomalies avec une certaine probabilité. Si une donnée est classée comme anomalie avec une probabilité supérieure à un certain seuil, l'IDS envoie une alerte

39

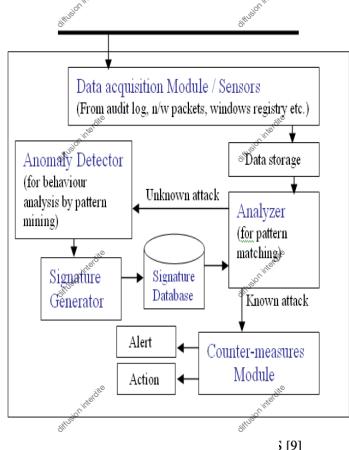
## TECHNIQUES DE DÉTECTION D'INTRUSIONS: PAR ANOMALIE

- Avantages:
  - Il permet de détecter des attaques zero-day
- Inconvénients:
  - Il peut échouer à détecter des attaques connues
  - Il peut aussi échouer à détecter des attaques nouvelles
  - Comment garantir que la dataset utilisée dans la phase d'apprentissage ne contient pas des attaques ?
  - Le taux de FP est souvent élevé, car des activités peuvent être différentes sans pour autant être malicieuses
- C'est une technique de détection largement discutée dans la Recherche, mais pas encore utilisée dans la pratique.

40

## TECHNIQUES DE DÉTECTION D'INTRUSIONS: ANOMALIE & SIGNATURE

- Combiner les 2 approches:



## TECHNIQUES DE DÉTECTION D'INTRUSIONS: PAR COMPORTEMENT

- La détection basée sur le comportement analyse les actions qui se déclenchent suite à une donnée qui arrive sur le réseau ou le système:
  - Au lieu d'analyser l'exploit, on analyse le résultat de l'exploit
  - Les comportements (ou les résultats) peuvent être analysés en utilisant la détection par signature (denylist), par spécification (allowlist), ou par anomalie.
- Exemples:
  - Si un certain input résulte en l'accès imprévu à des fichiers critiques. Il y a probablement une attaque en cours
  - Si un certain input résulte en l'appel d'une fonction exec dans un programme C qui n'utilise pas une telle fonction. Il y a probablement une attaque en cours

42

## TECHNIQUES DE DÉTECTION D'INTRUSIONS: PAR COMPORTEMENT

- Avantages:
  - Il hérite les avantages de la technique utilisée pour décider si un comportement est normal ou anormal:
  - Il peut réaliser un taux faible de FP si on spécifie les comportements qui apparaissent rarement dans le fonctionnement normal d'un système ou programme (denylist), ou si on spécifie tous les comportements normaux (allowlist)
  - Il peut détecter des attaques zero-day (anomaly detection)
- Inconvénients:
  - L'IDS alerte après que l'attaque soit réalisée et réussie
  - Il peut générer des FN si le comportement de l'attaque se ressemble à un comportement normal.

43

## AUTRES TECHNIQUES DE DÉTECTION D'INTRUSIONS

- PenTesting:
  - L'idée consiste dans le fait d'executer délibérément des attaques contre le système qu'on cherche à protéger, au lieu de détecter passivement des intrusions. Et ensuite corriger les vulnérabilités qui ont été exploitées dans les attaques réussies.
- Avantages:
  - Il permet de détecter des attaques réelles avant leur réalisation
- Inconvénient:
  - Peut prendre du temps et de l'argent
  - N'est pas utile pour les systèmes qu'on ne peut pas modifier
  - Il peut être perturbateur: on ne sait pas à l'avance quelles seront les conséquences d'une attaque réussie.

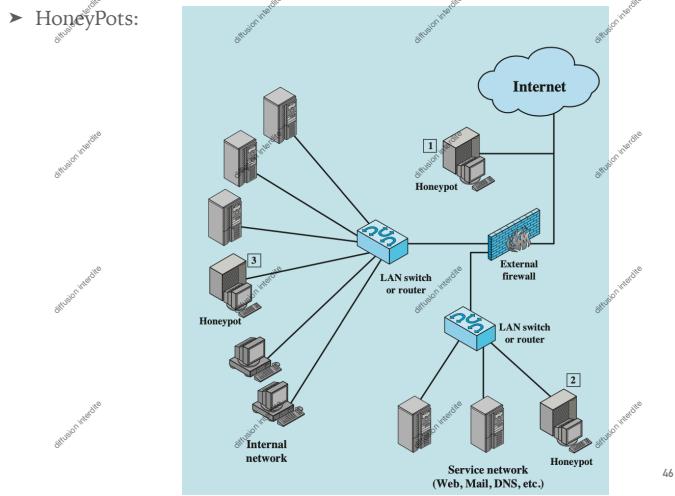
44

## AUTRES TECHNIQUES DE DÉTECTION D'INTRUSIONS

- HoneyPots:
  - Systèmes leurre conçus pour:
    - attirer un attaquant potentiel loin des systèmes critiques
    - collecter des informations sur l'activité de l'attaquant
  - Ils sont remplis d'informations qu'un utilisateur légitime du système n'accéderait pas
  - Ils sont une ressource sans aucune valeur pour la production:
    - recevoir des communications entrantes qui sont très probablement des scans ou des attaques
    - toute communication sortante suggère que le système a probablement été compromis
  - Exemple: Créer une @Mail qui n'est jamais utilisée par les utilisateurs légitimes. Si vous recevez des E-mail, c'est que ce sont des spams

45

## AUTRES TECHNIQUES DE DÉTECTION D'INTRUSIONS



46

## AUTRES TECHNIQUES DE DÉTECTION D'INTRUSIONS

- HoneyPots:
  - Avantages:
    - Il peut détecter des attaques zero-day
    - Il peut analyser les actions et la stratégie des attaquants
    - Il peut détourner les attaquant des vrais systèmes critiques
  - Inconvénients:
    - Il peut être difficile de convaincre les attaques à interagir avec un HoneyPot
    - Il faut donc beaucoup de temps et d'efforts pour créer un HoneyPot convaincant.

47

## AUTRES TECHNIQUES DE DÉTECTION D'INTRUSIONS

- Forensics:
  - C'est un complément essentiel à la détection d'intrusions. Il consiste à analyser ce qui s'est passé après une execution réussie d'une attaque dans le but de comprendre:
    - Pourquoi l'attaque a-t-elle réussi ?
    - Quelles défenses il faut mettre en place pour ne pas être attaqué à nouveau ?
    - Quelles vulnérabilités il faut corriger ?
    - Comment recouvrir l'état du système avant attaque ? (reprise de services, restauration de données,..)
  - C'est un sujet à part entière qu'il faut étudier.

48

## AUTRES TECHNIQUES DE DÉTECTION D'INTRUSIONS

- IPS (Intrusion Prevention System):
  - diffusion interne
  - diffusion interne
  - diffusion interne
  - diffusion interne
  - diffusion interne
- C'est un IDS mais qui bloque aussi les attaques. Il est souvent plus utilisé que l'IDS:
  - Il peut changer dynamiquement les règles de Firewall
  - Il peut forger des paquets pour stopper l'attaque
- Inconvénients:
  - Il ne peut pas être utilisé avec une analyse rétrospective.
  - Il ne peut pas être utilisé s'il est placé off-path.
  - Les FP sont coûteux car on affecterait des utilisateurs légitimes

49

## LES ATTAQUES SUR LES IDS

- L'IDS est aussi un système avec des ressources limitées, donc il peut être victime à des attaques DoS:
  - diffusion interne
  - diffusion interne
  - diffusion interne
  - diffusion interne
  - diffusion interne
- L'IDS analyse des codes qui peuvent être malicieux, donc il peut être victime d'une injection de code:
  - diffusion interne
  - diffusion interne
  - diffusion interne
  - diffusion interne
  - diffusion interne
- L'IDS exécute un code non fiable pour savoir s'il est malicieux, alors il se fait attaquer --> Utiliser des environnements isolés (SandBox, Emulation..)
  - diffusion interne
  - diffusion interne
  - diffusion interne
  - diffusion interne
  - diffusion interne

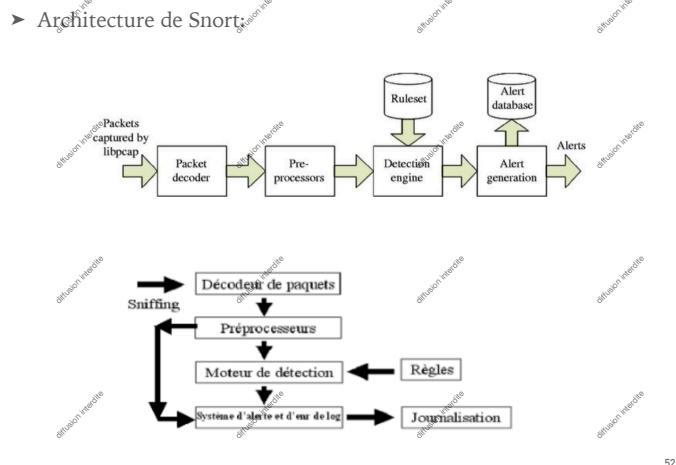
50

## EXEMPLE D'UN IDS/IPS: SNORT

- Snort est un IDS/IPS open-source créé par Martin Roesch en 1998. Il est maintenant maintenu par Cisco et il reste l'un des outils de détection d'intrusion les plus utilisés dans le monde.
- Snort utilise la technique de détection basée sur la signature
- Il fonctionne en analyse de paquets réseau en temps réel (NIPPS) pour:
  - La détection des intrusions
  - La détection des extrusions
  - Le filtrage de paquets (IPS)
- Il peut aussi fonctionner en mode:
  - Sniffer: il écoute passivement le réseau et affiche les paquet en temps réel (à l'instar tcpdump)
  - Logger: il enregistre les paquets dans des fichiers sur le disque

51

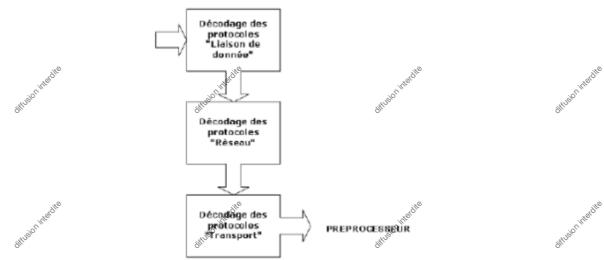
## EXEMPLE D'UN IDS/IPS: SNORT



52

## EXEMPLE D'UN IDS/IPS: SNORT

- Architecture de Snort: Le décodeur de paquets
- Il reçoit les trames brutes depuis libpcap
- Il transforme les éléments des protocoles (couche2->couche4) en une structure de données interne exploitable par Snort.
- Il détecte des erreurs basiques (syntaxiques)



53

## EXEMPLE D'UN IDS/IPS: SNORT

- Architecture de Snort: Le pré-processeur
- Il est lancé après le décodage de paquets
- Il ré-assemble les paquets IP fragmentés, il reconstruit le flux TCP, et il fait de la normalisation applicative (ex. enlève les encodages bizarres)
- Il fait un premier filtrage qui permet de détecter certaines intrusions:
  - Couche 2: ARP Spoofing..
  - Couche 3: Tiny Fragment, Fragment overlap..
  - Couche 4: Scan de ports, SYN Flood...
  - Couche 7: Path Traversal, injection de commandes...
- Si une intrusion est détectée à ce niveau, une alerte est envoyée directement au système d'alertes.

54

## EXEMPLE D'UN IDS/IPS: SNORT

- Architecture de Snort: Le moteur de détection
- Il utilise des règles Snort pour faire la détection des intrusions. Si un paquet correspond à une règle alors l'action définie dans la règle est exécutée:

Action	Description
alert	Generate an alert using the selected alert method, and then log the packet.
log	Log the packet.
pass	Ignore the packet.
activate	Alert and then turn on another dynamic rule.
dynamic	Remain idle until activated by an activate rule , then act as a log rule.
drop	Make iptables drop the packet and log the packet.
reject	Make iptables drop the packet, log it, and then send a TCP reset if the protocol is TCP or an ICMP port unreachable message if the protocol is UDP.
sdrop	Make iptables drop the packet but does not log it.

55

## EXEMPLE D'UN IDS/IPS: SNORT

- Architecture de Snort: Le moteur de détection
- Snort utilise un langage simple et flexible pour définir les règles
- Chaque règle se compose d'une entête et zéro ou plus d'options
- Les règles sont groupées en plusieurs catégories sous forme de fichiers.
- Snort vient avec un ensemble de règles prédefini. Elles ne sont pas activées automatiquement, il faut les activer dans le fichier de configuration snort.conf
- Il est aussi possible de définir de nouvelles règles dans un fichier séparé et de les activer dans snort.conf

```
include $RULE_PATH/local.rules
include $RULE_PATH/bad-traffic.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/scan.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
include $RULE_PATH/telnet.rules
include $RULE_PATH/rpc.rules
include $RULE_PATH/rservices.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/tftp.rules
```

## EXEMPLE D'UN IDS/IPS: SNORT

- Architecture de Snort: Le moteur de détection
- L'entête d'une règle Snort:
- <action> <protocole> <IP\_source> <port\_source> -> <IP\_destination> <port\_destination> (options)
- Action: alert, log, drop, sdrop, reject..
- Protocole: IP, TCP, UDP, ICMP,
- IP\_source, port\_source: ex. 192.168.1.0/24 , [80,443]
- IP\_destination, port\_destination: ex any,any
- Flèche: sens du trafic, soit unidirectionnel ou bidirectionnel
- Exemple:
- alert tcp any any -> 192.168.1.10 80 signifie que Snort va surveiller tout le trafic web vers le serveur 192.168.1.10

57

## EXEMPLE D'UN IDS/IPS: SNORT

- Architecture de Snort: Le moteur de détection
- Les Options d'une règle Snort:
- <action> <protocole> <IP\_source> <port\_source> -> <IP\_destination> <port\_destination> (options)
- Les options sont une suite de paires clé:valeur séparées par ; qui permettent de raffiner les règles Snort
- Exemples:
  - msg:"Texte descriptif" → message dans l'alerte
  - sid:10001; → identifiant unique de la règle
  - content:"abc"; → recherche d'une chaîne dans le payload
  - nocase; → insensible à la casse
  - pcre:"/regex/"; → expression régulière
  - priority:1; → priorité de l'alerte
  - dsizet:100; → taille du payload
  - ttl:< 32; → valeur du TTL
  - minfrag: 128; → fixe un seuil minimal pour la taille d'un paquet fragmenté

58

## EXEMPLE D'UN IDS/IPS: SNORT

- Architecture de Snort: Le moteur de détection
- Exemples de règles Snort:
- alert icmp any any -> any any (msg:"ICMP Echo Request Detected"; sid:10001;)
- alert tcp any any -> 192.168.1.10 80 (msg:"SQL Injection attempt"; content:" OR 1=1"; nocase; sid:10002;)
- drop tcp any any -> any 22 (msg:"Attempt to access SSH port 22"; sid:10003;)
- alert tcp any any -> 192.168.1.0/24 any (flags:SF; msg:"Possi. SYN FIN scan";sid:10004;)
- alert tcp any any -> 192.168.1.0/24 any (msg:"NMAP TCP SYN Scan Detected"; flags:S; sid:10005;)
- ...

59

## EXEMPLE D'UN IDS/IPS: SNORT

- Architecture de Snort: Système des alertes et enregistrement des logs
- Il s'occupe de l'enregistrement des logs et de la génération des alertes.
- Les logs sont par défaut stockés dans un répertoire spécifique /var/log/snort (ou C:\Snort\log selon l'OS)
- Il est possible de consulter les alertes soit dans les fichiers de logs soit via une solution SIEM (ex. Wazuh).

60

## EXEMPLE D'UN IDS/IPS: SNORT

- Architecture de Snort: Système des alertes et enregistrement des logs
- Format des alertes:
  - Fast: rapide, une seule ligne par alerte (isible par un humain)
  - Full: alerte détaillée avec toutes les infos du paquet
  - Unified: format binaire, optimisé pour être lu par un collecteur externe (ex. SIEM)
  - Syslog: envoie les alertes dans /var/log/syslog (ou /var/log/messages selon OS)

```
[**] [1:10001:1] ICMP Echo Request Detected [**]
```

```
[Priority: 0]
```

```
01/01-12:45:33.123456 192.168.1.5 -> 192.168.1.10
```

```
ICMP TTL:64 TOS:0x0 ID:54321 IpLen:20 DgmLen:84
```

```
Type:8 Code:0 ID:12345 Seq:1 ECHO
```

61

## EXEMPLE D'UN IDS/IPS: SNORT

- Avantages:
  - Ne fait pas tout, mais le fait correctement
  - N'engendre pas un ralentissement notable du trafic
  - Simplicité d'écriture des règles
  - Richesse de la base de signatures
  - Logiciel open-source: sources et signatures accessibles
  - Installation simple et rapide
  - Portable
- Limites:
  - Ne détecte pas tout
  - Mise à jour et ajout régulières des signatures
  - L'écriture et la vérification de règles peut demander du temps et de l'effort.

62

## EXEMPLE D'UN IDS/IPS

- Environnements sans fil (Wi-Fi):
- Utiliser des IDS "filaires" augmentés de signatures d'attaques contre la couche de niveau 2 du réseau sans-fil (802.11)
- Ou bien utiliser WIDS: IDS dédiés sans fil avec interface en mode Monitor.
- Exemples: Aruba Networks (Wireless Intrusion Protection Module), AdventNet, AirDefense Enterprise

63

## RÉFÉRENCES

- Transparents de Anas Abou El Kalam
- Transparents Université de Berkeley
- De la documentation en ligne

64