Retour d'expérience sur Prelude

OSSIR Paris / 11 février 2014

Mathieu Mauger – Consultant Sécurité (Mathieu.Mauger@intrinsec.com)
Guillaume Lopes – Consultant Sécurité (Guillaume.Lopes@Intrinsec.com)

@Intrinsec_Secu

Plan

Introduction

- Mathieu Mauger
  - → Consultant sécurité SI
    - ✓ Tests d'intrusion
- Guillaume Lopes
  - → Consultant sécurité SI
    - ✓ Tests d'intrusion
    - ✓ Audits (architecture, configuration et organisationnel)

- Intrinsec : acteur historique de la sécurité des SI (1995)
  - → Hébergement et infogérance des SI
  - → Sécurité de l'information
    - ✓ Pentest / Audit / Conseil / SOC / CERT

Introduction

- Prelude
  - Créé en 1988 par Yoan Vandoorselaere
  - Première fonction : IDS
  - Évolution vers une solution de type SIEM
  - Rachat de la solution par C-S en 2012

- Trois licences
  - OSS : version communautaire sous licence GPLv2 fournissant les fonctionnalités de base
  - Professionnelle : Intégration de fonctions supplémentaires
    - ✓ Ticketing, authentification LDAP, génération de rapports, etc.
  - Entreprise : Ajout de fonctionnalités supplémentaires permettant la mise en place d'un SOC selon l'éditeur

Introduction

Pourquoi avoir utilisé Prelude ?

Splunk

- ✓ Limitation à 500 MB d'évènements par jour
- ✓ Surveillance et alerte des évènements limitées
- ✓ Solution propriétaire

Alienvault OSSIM

- ✓ Système d'exploitation imposé (Debian)
- ✓ Architecture monolithique (sur la version communautaire)
- ✓ Intégration de nombreux outils non nécessaires dans notre contexte
  - Compliance / PCI-DSS / etc.

Plan

- Introduction
- Infrastructure
- Composants
- Alerting
- Statistiques
- Cas d'utilisation
- Conclusion

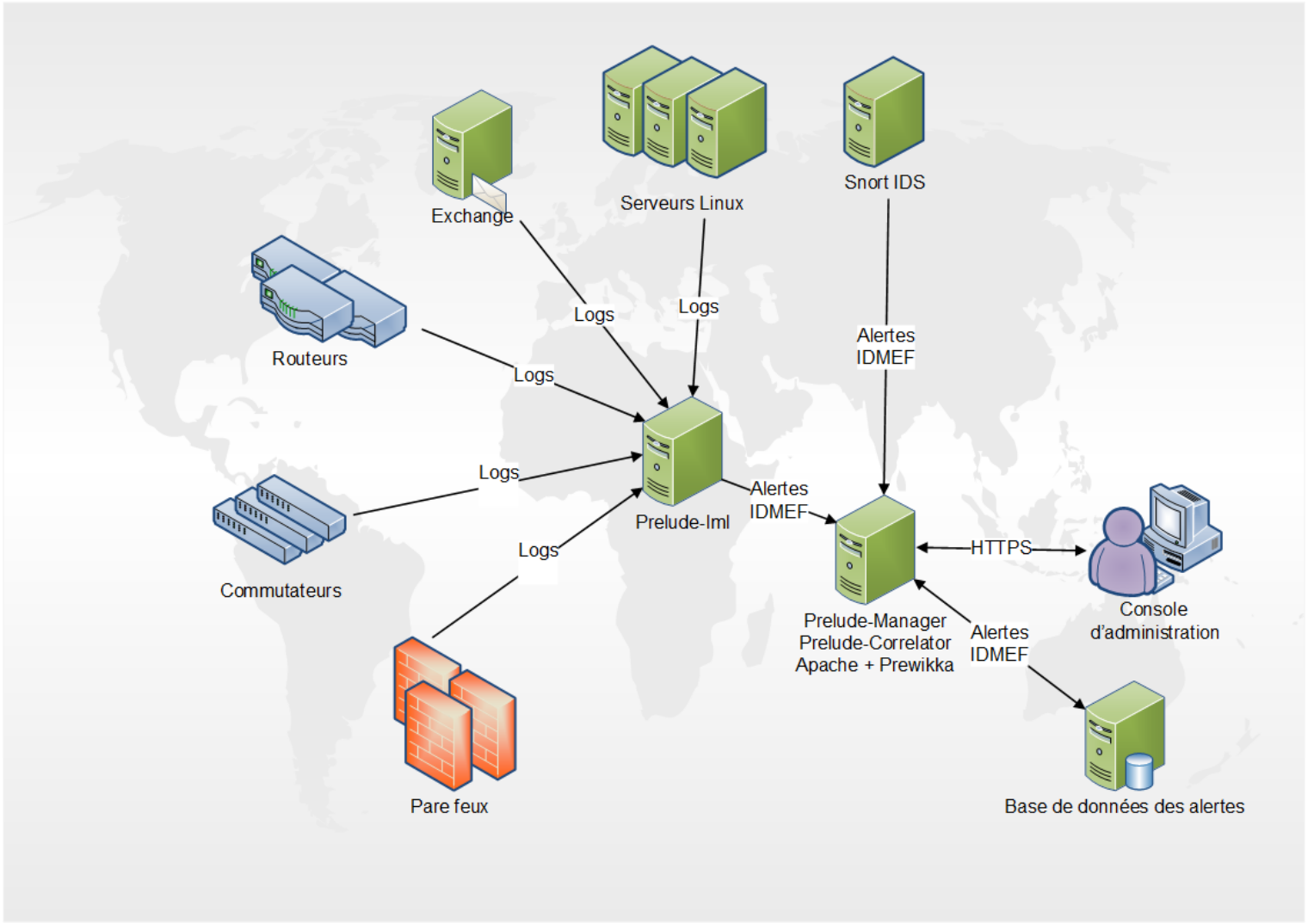
Infrastructure

- Architecture
  - Modèle distribué : Un manageur contrôlant diverses sondes
    - ✓ Prelude-lml, prelude-correlator, Snort, etc.
  - Communications chiffrées avec une clé RSA 2048 bits puis à l'aide d'un certificat x509
  - Utilisation du langage IDMEF (RFC 4765) pour le dialogue inter-équipements

Infrastructure

- Équipements supervisés
  - 3 routeurs linux Kernel 2.6
  - 2 pare-feu linux (Iptables)
  - 100 serveurs linux
  - 2 serveurs Windows 2008R2 (Exchange et serveur de fichiers)
  - Une sonde Snort
  - Un commutateur Cisco
- Exclusion des équipements utilisateurs (inconsistance des logs)
- Équipements de la solution
  - Un serveur de centralisation de logs
    - ✓ 8 GB RAM, Processeur XEON E7, 500 GB d'espace disque
  - Un serveur maitre de management
    - ✓ 16 GB RAM, Processeur XEON E7, 250 GB d'espace disque

Infrastructure



Plan

- Introduction
- Infrastructure
- Composants
- Alerting
- Statistiques
- Cas d'utilisation
- Conclusion

Composants

- Modèle distribué
  - Bibliothèque Prelude : Permet la compatibilité et la communication entre les différents équipements de la solution
  - Manager : Reçoit et organise l'ensemble des alarmes
  - Prelude-lml : Collecte les évènements de sécurité dans les logs et les transforme en alerte (IDMEF)
  - Prelude-correlator : Corrélation des alertes afin de regrouper des alertes ou définir des schémas d'attaques
  - Autres sondes : Snort, OSSEC, Firewall ASA, etc. : Certains équipements intègrent nativement la bibliothèque Prelude
  - Prewikka : Interface de visualisation des alertes
- Il est nécessaire de centraliser les fichiers d'évènements des équipements non compatibles (Windows 7, FreeBSD, Apache, Samba, etc.)

Composants

- Bibliothèque Prelude
  - Interface unique et standard de communication de la solution
  - Interopérabilité par l'utilisation du format IDMEF
  - Gestion de l'authentification et du chiffrement des communications
  - Nécessaire sur chaque sonde

- Attention : Par défaut, l'intervalle des Heartbeats défini par la bibliothèque est trop faible

Composants

- Prelude-Manager
  - → Réception et stockage des alertes
    - ✓ Fichiers plats : Format texte
    - ✓ Fichier XML-IDMEF : Stockage suivant le standard RFC pour l'import et l'export de données vers les services Prelude
    - ✓ SQL : Insertion des alertes dans une base de données pour le traitement par des logiciels tiers
  - → Intégration des règles de pré-filtrages au niveau des alertes reçues
- Un manager peut être l'esclave d'un autre manager
  - → Utile pour la gestion de plusieurs sites distants

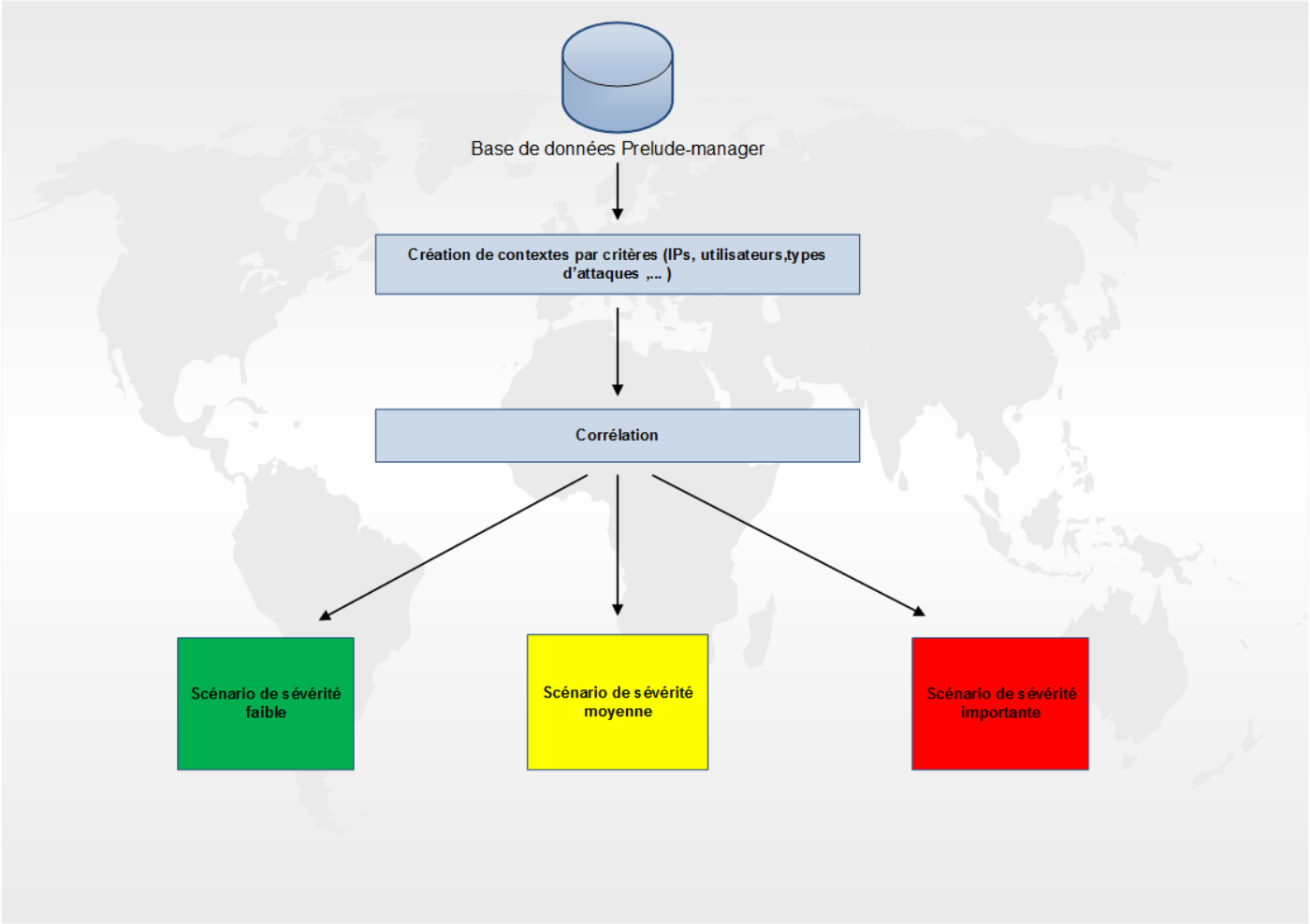
Composants

- Prelude-lml
  - Deux méthodes d'analyse
    - ✓ Analyse des fichiers de logs reçus en local
    - ✓ Récupération des logs par UDP
  - Compatibilité native avec de nombreux logiciels
  - Utilisation de PCRE pour la récupération des évènements avant la transformation au format IDMEF
  - Utilisation du format SYSLOG par défaut
  - Possibilité de modifier les règles d'analyses pour prendre en compte les formats exotiques

Composants

- Prelude-Correlator
  - Récupération des alertes directement dans la base de données
  - Regroupements des alertes selon des critères prédéfinis
  - Mise en corrélation, périodiquement, des alertes issues d'une précédente corrélation
  - Intégration de modules de corrélation par défaut :
    - ✓ OpenSSHAuth : Gestion des alertes relative à OpenSSH
    - ✓ EventStorm : Utilisé dans le regroupement d'un nombre important d'alertes (e.g DDoS)
    - ✓ Firewall : Corrélation des alertes de pare-feu
  - La création de nouvelles règles est simplifiée par des classes Python préexistantes

Composants



Composants

- Prewikka
  - Solution officielle du projet Prelude depuis la disparition des interfaces PHP et Perl
  - Trois fonctionnalités majeures
    - ✓ Gestion des « heartbeats »
    - ✓ Exportation et visualisation des tableaux de bord graphiques
    - ✓ Règles de filtrage à la volée
  - Différences importantes d'ergonomie/fonctionnalités entre la version communautaire et la version entreprise

Plan

- Introduction
- Infrastructure
- Composants
- **Alerting**
- Statistiques
- Cas d'utilisation
- Conclusion

Alerting

- La fonctionnalité de filtrage des alertes est intégrée au manager Prelude
  - → Les alertes sont filtrées avant leur insertion en base de données
  - → Deux systèmes de filtrage
    - ✓ Utilisation de règles simples reposant sur la définition de trois paramètres
      - *Name* : Nom de la règle
      - *Rule* : Le type d'évènement, au format IDMEF, qui sera traité par le filtre (ex. élévation de privilèges)
      - *Hook* : L'action à effectuer avec les alertes qui sont concernées par le filtrage
    - ✓ Affinage du traitement avec l'instruction thresholding
      - *Count* : Définit la limite de prise en compte pour un évènement
      - *Limit* : Temps durant lequel le paramètre count est actif
      - *Path* : Permet d'augmenter la précision du filtrage
      - *Hook* : Comportement similaire au système de filtres précédent

Alerting

- Exemple

```
[idmef-criteria=BruteForce]
rule = alert.assessment.impact.severity == 'Brute Force attack'
hook = thresholding[BruteForce]

[thresholding=BruteForce]
path = alert.classification.text, alert.target(0).node.address(0).address
count = 1
limit = 3600
hook = smtp[test]
```

- Traduction
  - Les attaques répertoriées comme bruteforce (rule) sont envoyées au filtre thresholding nommé BruteForce (hook)
  - Les évènements entrants provenant de la même adresse (path) sont remontés une fois (count) toutes les heures (limit) au travers d'un email (hook)

Alerting

- Gestion de la remontée des alertes par trois plugins intégrés au Manager
  - Mail
    - ✓ Avantage : Peu de latence entre l'enregistrement de l'évènement et la réponse des équipes
    - ✓ Inconvénient : Effet « mail bombing » si les règles ne sont pas correctement effectuées
  - Fichiers texte
    - ✓ Avantage : Résultat brut au format IDMEF
    - ✓ Inconvénient : L'extraction des données est difficile
  - XML
    - ✓ Avantage : Exportation et analyse du contenu facilitées
    - ✓ Inconvénient : Volume important d'espace disque utilisé

Plan

- Introduction
- Infrastructure
- Composants
- Alerting
- **Statistiques**
- Cas d'utilisation
- Conclusion

Statistiques

- Volume d'alertes
  - → 80 GB par trimestre pendant la phase de pré-production
    - ✓ Filtres sommaires
    - ✓ ~ 60% d'alertes non pertinentes
  - → 35 GB pendant la phase de production après l'application des filtres
    - ✓ Utilisation de l'empilement des filtres
    - ✓ ~ 15 % d'alertes non pertinentes

Statistiques

- Des alertes pas toujours utiles
  - ✓ Scans de vulnérabilités
    - **30% pré-production vs 20% production**
  - ✓ Attaques de type bruteforce sur les services accessibles depuis Internet
    - **60% pré-production vs 35 % production**
  - ✓ Attaques ciblées (tentatives d'exploitation de vulnérabilités)
    - **5% pré-production / 5% production**
    - Script Kiddies
    - Le grand méchant chinois ;)
  - ✓ Dysfonctionnements internes et atteintes à la sécurité
    - **5% pré-production / 40% production**
    - Boucles réseaux
    - Rattachement de deux sous réseaux isolés
    - SSH root
    - ...

Statistiques

- 70% des évènements de sécurité ne peuvent être corrigés aisément
  - Contraintes techniques
  - Services accessibles depuis Internet indispensables
  - Facteurs humains

Plan

- Introduction
- Infrastructure
- Composants
- Alerting
- Statistiques
- Cas d'utilisation
- Conclusion

Cas d'utilisation

- Un brute force SSH vs connexion légitime SSH sur le compte root
  - → Le brute force va générer une alerte majeure de type brute force puis une alerte de niveau critique si la connexion SSH est réussie
  - → Dans le cas d'une connexion SSH légitime, une seule alerte de niveau majeure va être générée
- Cette corrélation peut être faite uniquement si une configuration préalable de l'outil a été effectuée
- Sans configuration, les deux authentifications seront classées comme majeures

Cas d'utilisation

- Détecter une boucle réseau et/ou une interface défectueuse sur un commutateur
  - → Emission d'alertes répétées indiquant la montée et l'arrêt d'une interface
  - → Indication du ou des interfaces réseau concernées par l'évènement
- Résolution rapide de l'incident
- Corrélation d'évènements minimes permettant d'éviter une perte de disponibilité

Cas d'utilisation

- Connexion non légitime de deux sous réseaux
  - Augmentation importante du nombre de paquets ARP
  - Alerte émise par les équipements agissant au niveau réseau
    - ✓ DNS, DHCP, Commutateur, etc.
  - La solution estime qu'une attaque de type MITM ou DoS est en cours et envoie une alerte à l'administrateur
  - Les adresses IP et ports de connexion renseignés dans l'alerte permettent d'identifier rapidement la source de l'incident
- Détection rapide du problème permettant d'éviter un cloisonnement défectueux

Plan

- Introduction
- Infrastructure
- Composants
- Alerting
- Statistiques
- Cas d'utilisation
- Conclusion

Conclusion

- Modèle d'infrastructure distribuée permettant l'intégration sur plusieurs sites et/ou sous-réseaux
- Système de filtre complet reposant sur la syntaxe IDMEF permettant d'affiner précisément les recherches et actions à mener
- Un nombre croissant d'équipements compatibles nativement avec la solution
  - → Windows, Snort, OSSEC, etc.
- Création de reporting visuels afin de mettre en avant les attaques et les besoins du SI en terme de sécurité (version entreprise ou développement personnel)

Conclusion

- La solution dépend entièrement des logs générés par les équipements du Système d'Information
  - → Vulnérable aux attaques de dissimulation classique
    - ✓ Arrêt des enregistrements des logs
    - ✓ Fragmentation IP
    - ✓ Évasion IDS
- Coût important en ressources
  - → Aspects humain (configuration, analyse des alertes, corrections, etc.)
  - → Équipements à même d'encaisser la charge
- Solution non rentable pour les petites structures

Conclusion

- Problématique
  - → Que faire des évènements récurrents ?
    - ✓ Scan de vulnérabilités
    - ✓ Brute force
- Il n'existe pas de solution miracle
  - → Supprimer les équipements cibles de la surveillance
    - ✓ Oui mais si une attaque réussie, on est aveugle
  - → Arrêter les services cibles
    - ✓ Perte de productivité, rentabilité, visibilité, etc.
  - → Ne plus utiliser les humains ? ☺

intrinsec
BUSINESS CONTINUITY AND SECURITY

Merci de votre attention
Questions ?