DE LA RECHERCHE À L'INDUSTRIE

cea

ensiie
Paris-Évry | Strasbourg

www.cea.fr

Architecture d'un système d'exploitation

Introduction à la Sécurité Informatique en Environnement HPC

frederic.combeau@cea.fr

Novembre 2022

Plan

- **La sécurité informatique**
  - Concepts : confidentialité, intégrité, disponibilité et non-répudiation
  - Méthodologie : menaces, vulnérabilités, risques, impacts, mesures, vérifications
- **Spécificités du HPC**
  - Utilisateurs virtuels, postes client, cloisonnement, surface d'attaque, puissance de calcul, traçabilité, disponibilité
- **Grands principes de sécurité**
  - Système minimal, tout interdire, séparation de privilèges, mises à jour, surveillance, sécurité en profondeur, architecture
- **Sécurité système**
  - Authentification, PAM, protections de fichiers, chroot, cgroups, su et sudo, compte captif et shell restreint, chiffrement, DAC vs MAC, SELinux
- **Sécurité réseau**
  - Pare-feu, proxy, VPN, gestion des services
- **Surveillance du système**
  - Syslogd, auditd, intégrité des logs, accounting et collectd, audits et test de pénétration
- **Exemples de menaces et d'attaques**
  - Usurpation d'identité, ip spoofing, buffer overflow, ingénierie sociale et phishing, remote exploit, rootkit, race condition, ransonware

La sécurité informatique
Concepts et méthodologie

Spécificités du HPC pour la sécurité
Grands principes de sécurité
Sécurité système
Sécurité réseau
Surveillance du système
Exemples de menaces et d'attaques

La sécurité informatique - Concepts

Confidentialité

- Usurpation d'identité
- Accès illicite à des données
- Ex-filtration de données

Disponibilité

- Consommations excessives de ressources (CPU, mémoire, disques, réseau), famine
- Surcharge d'un service

Intégrité

- Modifications et/ou effacements de données, de codes, de bibliothèques
- Rebond vers des ressources externes aux centre de calcul

Non-répudiation

- Atteinte à la comptabilité et/ou à l'imputation
- Compromettre la surveillance du centre de calcul
- S'affranchir de la traçabilité

La sécurité informatique - Méthodologie

Tous les éléments utilisés dans la méthodologie sont des atteintes aux propriétés de base de la sécurité informatique : confidentialité, intégrité, disponibilité et non-répudiation

Menaces

- **Une menace utilise à son profit une faille (vulnérabilité) dans un système d'information, afin d'y pénétrer, ce qui aura des conséquences (impact)**
- **Une menace peut être assimilée à un attaquant, mais une inondation peut être une menace**

Vulnérabilités

- **Une vulnérabilité (technique ou humaine) est une faille, une faiblesse ou un défaut permettant de faire faire une action non prévue par un logiciel ou un acteur humain**
- **Une vulnérabilité est associé à une perte de contrôle (logiciel, flux de commandes, flux de données, procédure, …)**

La sécurité informatique - Méthodologie

Risques

- Un risque est une probabilité et/ou possibilité qu'une menace donnée exploite le ou les vulnérabilités d'un acteur et nuise au système d'information
- Un risque est caractérisé par une conséquence (l'impact) et une probabilité et/ou possibilité
- Une analyse de risque va essayer de caractériser les menaces associées à la probabilité de leurs occurrences pour en évaluer, dans un second temps, les impacts

Impacts

- L'impact est la conséquence de l'exploitation d'une vulnérabilité par une menace
- L'impact doit être évalué afin de définir si le risque d'une menace est acceptable ou non et de pouvoir prioriser les mesures à mettre en place pour atténuer le risque, la menace ou la vulnérabilité

La sécurité informatique - Méthodologie

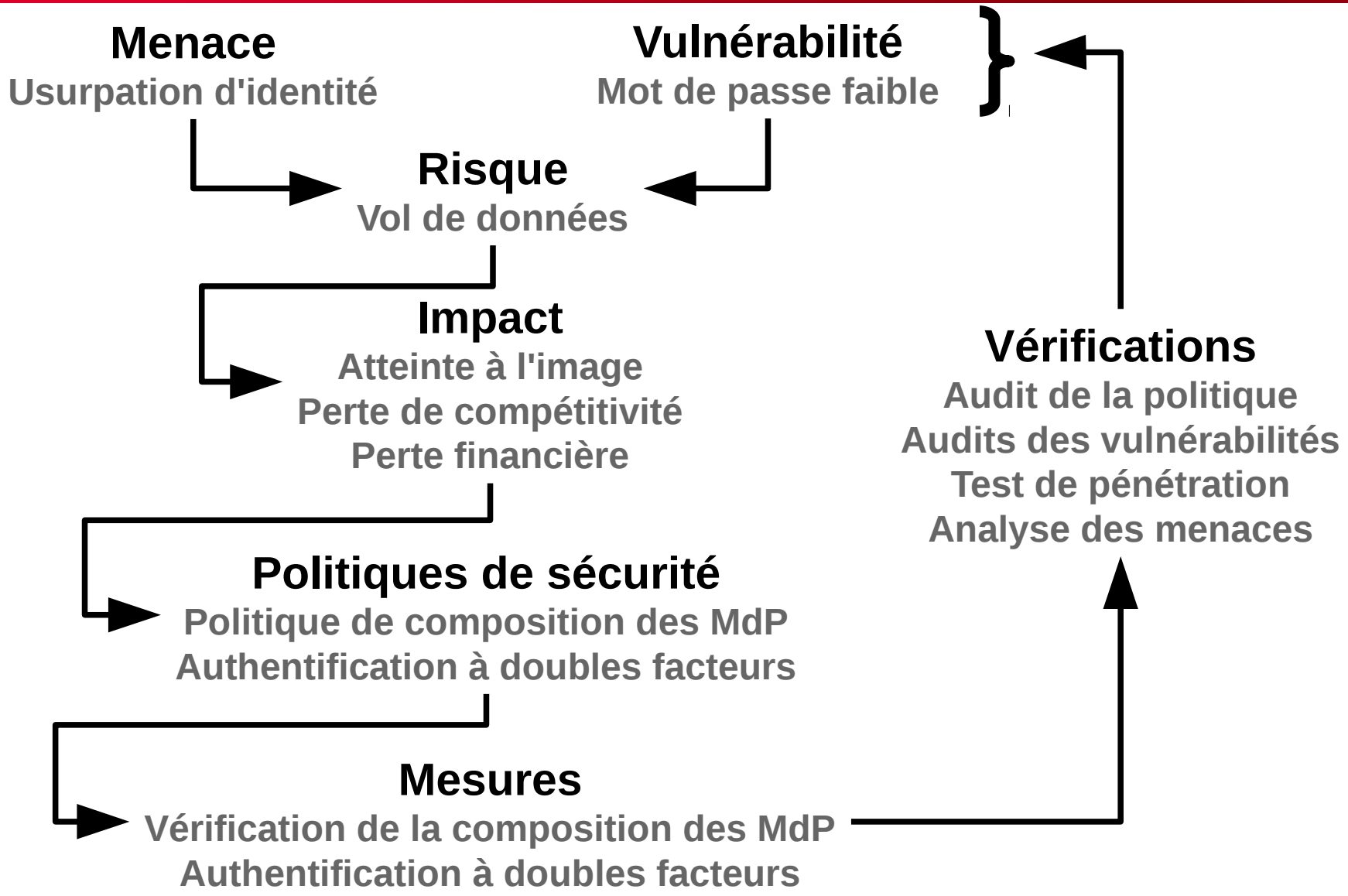
Mesures

- Une mesure est un élément technique ou humain afin d'agir pour limiter l'impact d'une menace
- Une mesure peut agir sur la menace (limiter le périmètre d'attaque), sur les vulnérabilités (application des patches, formation des utilisateurs) ou sur l'impact (surveiller pour détecter et pouvoir répondre au plus vite)
- Selon l'analyse de risque et les impacts identifiés, une politique de sécurité doit être définie et appliquée sur un SI par des mesures avec des priorités (roadmap)

Vérifications

- Les menaces et les vulnérabilités étant en constante évolution, la méthodologie doit être constamment ré-itérée afin de les prendre en compte pour faire évoluer la politique de sécurité
- Une vérification des mesures mises en place est nécessaire pour s'assurer que la politique de sécurité est bien en place

La sécurité informatique - Méthodologie



La sécurité informatique

Spécificités du HPC pour la sécurité

Utilisateurs virtuels, postes client, cloisonnement, surface d'attaque, puissance de calcul, traçabilité, disponibilité

Grands principes de sécurité
Sécurité système
Sécurité réseau
Surveillance du système
Exemples de menaces et d'attaques

Spécificités du HPC pour la sécurité

Quelles sont les spécificités en termes de sécurité informatique dans le monde HPC ?

- **Utilisateur virtuel**
  - Pas de connexions physiques à un calculateur ou à un centre de calcul (protection physique des éléments informatiques)
  - Connexions réseau comme point d'entrée (sécurité de la connexion, bande passante réseau)

- **Poste client non maîtrisé**
  - Connexion à partir de poste de travail non maîtrisé (sécurité)
  - Connexion à partir de réseau non maîtrisé (sécurité)

- **Surface d'attaque**
  - Surface d'attaque de type réseau et de niveau Internet
  - Accès Internet en entrée comme potentiellement en sortie
  - Assuré une bonne bande passante (débit et latence) tout en limitant l'accès uniquement depuis les sites légitimes

Spécificités du HPC pour la sécurité

Quelles sont les spécificités en termes de sécurité dans le HPC ?

- **Cloisonnement**
  - Gestion de plusieurs populations d'utilisateurs qui partagent un ou plusieurs calculateurs (performance des calculs) tout en préservant la confidentialité des données

- **Puissance de calcul disponible**
  - Attire des utilisateurs malveillants (brute-force de mots de passe, mining de BitCoins, déni de service distribué, disponibilité des moyens informatiques, taille importante du stockage)
  - Impact minimal de la sécurité sur les performances

- **Disponibilité**
  - Élément très important du HPC (souvent contractuel) pour ses différentes composantes : calcul, stockage, réseau

Spécificités du HPC pour la sécurité

Quelles sont les spécificités en termes de sécurité dans le HPC ?

- **Traçabilité**
  - Pas seulement au niveau sécurité
  - Accounting (facturation des ressources utilisées)
  - Surveillance (collecte et traitement des événements) et supervision (actions automatiques en fonction de certains éléments) du centre de calcul
  - Détection au niveau sécurité (avec ou sans réponse)
  - Masse de données importante à collecter et à traiter (data mining)

La sécurité informatique
Spécificités du HPC pour la sécurité

Grands principes de sécurité

Système minimal, tout interdire, séparation des privilèges, mises à jour, surveillance, sécurité en profondeur, architecture

Sécurité système
Sécurité réseau
Surveillance du système
Exemples de menaces et d'attaques

Grands principes de sécurité – Système minimal

Pourquoi installer et utiliser un système minimal ?

- Occupation inutile de l'espace disque
- Plus il y a d'applications, plus il a de vulnérabilités potentielles et plus il faudra appliquer les mises à jour (patches)
- Plus il y a d'applications, plus un intrus aura d'outils pour faciliter son intrusion
- Plus il y a de services, plus il est difficile de cloisonner le réseau
- Plus il y a de services, plus il est difficile d'atteindre un niveau élevé de sécurité
  - niveau de sécurité d'un système = niveau de sécurité du composants le moins sécurisé
- Difficulté de détecter un outil suspect parmi beaucoup d'applications
- Difficulté de contrôler l'intégrité de beaucoup d'applications et de services

⇒ Limiter les outils installés et les services actifs au strict nécessaire (à appliquer pour tous les éléments du SI)

Grands principes de sécurité – Tout interdire

En sécurité informatique, tout doit être interdit par défaut

- Afin de respecter le principe de n'autoriser que le strict nécessaire, tout est interdit par défaut
- Puis on autorise uniquement ce dont on a besoin
- Cela se concrétise, dans les configurations, par :
  - Les règles qui ne font qu'autoriser en début de configuration
  - Suivies d'une règle qui interdit tout, à la fin de la configuration
- Cette façon de faire permet de s'assurer que le ou les cas non prévus seront toujours couvert par la règle par défaut qui interdit tout
- **Attention de ne pas laisser une règle qui autorise tout dans la configuration (surtout après une phase de mise en place ou de débogage)**

Grands principes de sécurité – Séparation des privilèges

Qu'est-ce que la séparation des privilèges ?

- Même philosophie que le système minimal appliquée aux privilèges
- Une application ou un service n'a pas besoin de tous les privilèges tout le temps de son exécution
- Les privilèges ne doivent être utilisés que lorsqu'on en a besoin

- Exemple :
  - Une application qui a besoin du privilège root pour lire un fichier n'en a besoin que lors de l'ouverture du fichier, ensuite elle peut l'abandonner
  - Si une vulnérabilités est trouvée dans cette application en dehors de l'utilisation du privilège, son impact est moindre qu'avec le privilège utilisé tout au long de l'exécution

⇨ Les phases critiques (utilisation des privilèges) de l'exécution de l'application sont donc mieux gérées

Grands principes de sécurité – Mises à jour

- Les mises à jour sont le premier principe de sécurité informatique
- Permet de limiter les vulnérabilités (en principe)
- Et sans vulnérabilité il n'y a pas de menace

- Difficulté d'appliquer les mises à jour :
  - Problème de disponibilité (les mises à jour peuvent casser l'intégration du produit ou certaines fonctionnalités) d'où la frilosité des administrateurs système pour les appliquer
  - Nécessité de configurer les services après leurs mises à jour
  - Nécessité de rebooter le système ou les services (mises à jour du noyau ou des bibliothèques)

- Exemple de Wanna Cry :
  - Exploitation en mai 2017 d'une vulnérabilité corrigée en mars 2017

Les mises à jour doivent être appliquées, suivies et contrôlées dans un délai raisonnable selon une politique de mises à jour

Grands principes de sécurité – Surveillance

- Tout système d'information doit être surveillé au niveau sécurité
- La surveillance permet de limiter l'impact d'une intrusion en permettant de la détecter et de prendre des mesures adéquates le plus vite possible
- Pour certaines vulnérabilités, la seule option est la détection pour déclencher des mesures souvent drastiques (isolation du système, coupure de l'accès réseau, …)

- La surveillance intègre la vérification de l'intégrité du système (modification des exécutables, des bibliothèques, du noyau, des fichiers de configuration, …)

- La surveillance se base sur les événements survenant sur le système d'information, ceux-ci doivent être collectés, stockés et analysés en s'assurant qu'ils n'ont pas été modifiés à chacune de ces étapes
  - un intrus essaie en premier lieu de laisser le minimum de traces pour ne pas être repéré

Grands principes de sécurité – Sécurité en profondeur

Qu'est-ce que le principe de sécurité en profondeur ?

- Il est plus efficace pour limiter une intrusion de placer une multitude de (petites) barrières sur tous les éléments du système d'information plutôt que d'avoir les systèmes de sécurité à un seul endroit
- Le principe de sécurité en profondeur permet de multiplier les mesures de sécurité avec différentes implémentations, à différents endroits et sur un maximum d'éléments

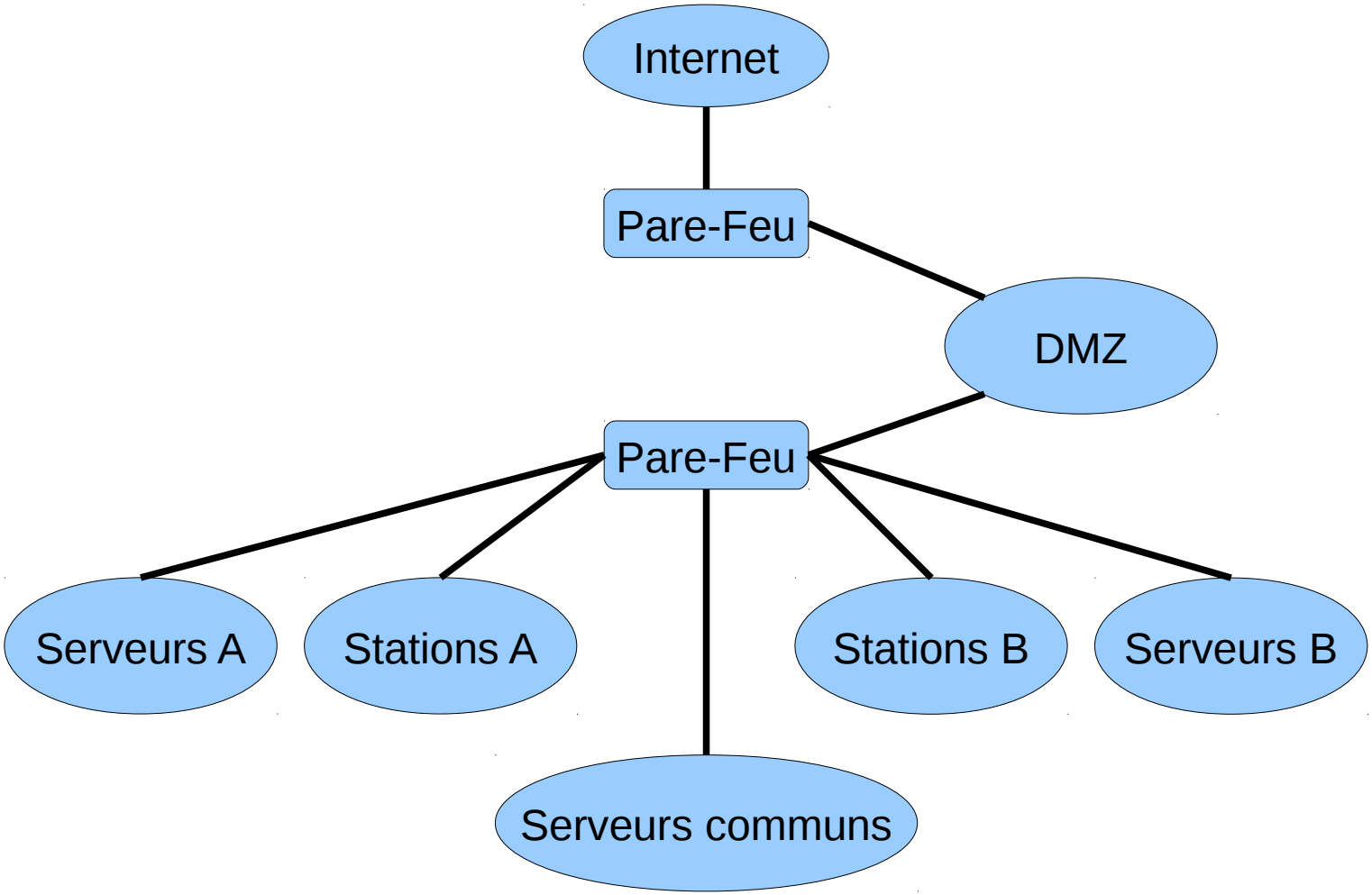
- Exemples :
  - L'utilisation d'un pare-feu central au niveau réseau ne dispense pas de définir des règles de contrôle de flux réseau sur chaque serveur
  - L'authentification à double facteur à l'entrée sur le réseau ne dispense pas d'authentifier les utilisateurs pour chaque service utilisé (LDAP, système de fichier réseau, Web interne)

Grands principes de sécurité – Architecture

- La sécurité est aussi implémenté au niveau de l'architecture du système d'information
- Un système à plat avec un pare-feu à l'entrée du réseau n'est pas une bonne architecture de sécurité
- Les différents éléments du système doivent être cloisonnés selon plusieurs critères
  - sensibilité, exposition, vulnérabilités résiduelles, criticité, …
- Le cloisonnement doit incorporer le principe de sécurité en profondeur et être réalisé par un ou plusieurs pare-feux ou des listes de contrôle d'accès (ACLs)
- Tous les services ne doivent pas être centralisés sur un seul serveur mais peuvent être regroupés dans une zone si leur sensibilité est équivalente

Grands principes de sécurité – Architecture

- **Exemple d'architecture :**



La sécurité informatique
Spécificités du HPC pour la sécurité
Grands principes de sécurité

Sécurité système

Authentification, PAM, protections fichiers, chroot, cgroups, su et sudo, compte captif et shell restreint, chiffrement du système de fichier, DAC vs MAC, SELinux

Sécurité réseau
Surveillance du système
Exemples de menaces et d'attaques

Sécurité système - Authentification

Qu'est-ce que l'authentification ?

- Tout système d'information moderne a besoin de connaître l'identité d'un utilisateur
- Pour cela il utilise deux méthodes successives
  - L'identification (login) et l'authentification (password)

- L'identification permet de déterminer qui est la personne
- L'authentification permet de prouver que l'on est bien la personne identifiée

- L'authentification peut être réalisée avec :
  - ce que l'on connaît (secret),
  - ce que l'on possède (token),
  - ce que l'on est (biométrie)
- L'usage de la biométrie pour l'authentification est controversé en France (il est plus vu comme un moyen d'identification) :
  - élément facilement copiable et impossible à révoquer

Sécurité système - Authentification

Qu'est-ce que l'authentification ?

- L'authentification à simple facteur n'utilise qu'un seul élément d'authentification
- L'authentification à double facteurs utilise deux éléments de types différents pour s'authentifier

- Exemples de méthodes d'authentification
  - Ce que l'on connaît : mot de passe statique, mot de passe dynamique (One Time Password), schéma géométrique, suite d'éléments graphiques
  - Ce que l'on possède : élément cryptographique comportant un secret : token, calculette, carte à puce pouvant être protégés par un mot de passe ou un code pin
    - Le secret permet de chiffrer un défi (challenge) pour s'assurer de sa possession ou permet de générer un mot de passe jetable et non rejouable

Sécurité système - Authentification

Qu'est-ce que l'authentification ?

- L'authentification peut être locale (mots de passe locaux à une machine) ou centralisée (serveur d'authentification) comme Radius, LDAP ou Kerberos

- L'authentification peut également être compatible avec une technologie de Single Sign On (SSO). Dans ce cas, la première authentification génère un jeton logiciel utilisé pour les authentifications à venir (transparent pour l'utilisateur)
  - Attention à la sécurité du jeton logiciel qui permet de prendre l'identité de l'utilisateur au niveau des services

Sécurité système - PAM

Pluggable Authentication Modules

- Permet de transférer la partie authentification d'une application dans les bibliothèques PAM
- C'est cette bibliothèque qui réalisera l'authentification
- L'application n'a plus à définir tous les types d'authentification, tout est pris en charge par la bibliothèque PAM que l'application doit appeler (et seulement celle-ci)
- L'authentification est totalement paramétrable à travers les PAM, application par application
- La grande majorité des système d'exploitation Unix utilise les PAM avec FreeBSD

Sécurité système - PAM

Pluggable Authentication Modules

- Chaque application possède sa configuration PAM dans le répertoire /etc/pam.d, dans un fichier portant son nom (par exemple /etc/pam.d/sshd)

- Le fichier de configuration a la syntaxe suivante <catégorie> <contrôle> <module_PAM_avec_ses_paramètres>

- 4 catégories différentes
  - auth : vérification de l'identité et de l'authentification
  - account : vérification des informations du compte (mot de passe expiré, appartenance à un groupe)
  - password : mise à jour du mot de passe (obligation de changer son mot de passe expiré)
  - session : préparation de l'utilisation du compte (log de la connexion, montage du répertoire utilisateur)

Pluggable Authentication Modules

- 4 contrôles différents
  - requisite : module obligatoire, une réponse négative met fin au traitement et renvoi NON OK
  - required : module obligatoire, une réponse négative ne met pas fin au traitement mais renverra NON OK
  - sufficient : module non obligatoire, une réponse positive arrête le traitement et renvoi OK (sauf si un module required précédant a renvoyé une réponse négative), une réponse négative n'est pas prise en compte
  - optional : module non obligatoire, la réponse est prise en compte si aucun autre module n'est présent
- Les PAM sont difficiles à maîtriser et peuvent conduire à de grosses erreurs de configuration (désactivation de l'authentification)
- Attention, avec les « requisite », de ne pas donner d'informations sur la configuration PAM

Pluggable Authentication Modules

- **Exemple (/etc/pam.d/login) :**

```
auth        required   pam_securetty.so
auth        required   pam_unix.so nullok
auth        sufficient pam_nologin.so
auth        required   pam_deny.so
account     required   pam_unix.so
password    required   pam_cracklib.so retry=3
password    required   pam_unix.so shadow nullok use_authtok
session     required   pam_unix.so
```

Rappels des droits Unix sur les fichiers

- Sous Unix, tout est fichier et tout fichier comporte des droits d'accès pour son utilisateur propriétaire, son groupe propriétaire et le reste du monde
- Ce sont les outils de base pour assurer le cloisonnement au sein du système de fichiers

- ls -l ./toto.sh donne

-rwxrw-r-- 1 frederic frederic 94 sept. 11 20:27 toto.sh

- Type de fichier
  - - (fichier classique)
  - d (répertoire)
  - l (lien souple)
  - c (périphérique de type caractère)
  - b (périphérique de type bloc)
  - p (pipe),
  - s (socket)

Sécurité système – Protections des fichiers

Rappels des droits Unix sur les fichiers

- Droits propriétaire
  - rwx (read, write, execute) + setuid (-)
- Droits groupe
  - rwx (read, write, execute) + setgid (- ou d)
- Droits autre
  - rwx (read, write, execute)+ sticky bit (- ou d)

- Fichier
  - setuid : exécution au nom du propriétaire du fichier
  - setgid : exécution au groupe du groupe du fichier
  - sticky bit : exécutable reste en mémoire

Rappels des droits Unix sur les fichiers

- Répertoire
  - r : lire les entrées d'un répertoire
  - w : modifier les entrées d'un répertoire (ajouter un fichier, effacer un fichier, changer un nom de fichier)
  - x : accéder au répertoire (cd)
  - setguid : tout fichier créé dans ce répertoire hérite du groupe du répertoire et tout répertoire créé hérite du setgid
  - sticky bit : interdit la suppression des fichiers dont on n'est pas propriétaire

- Notation octale
  - 0755 = rwxr-xr-x, 0640 = rw-r-----
  - 4640 = rwSr-----, 4740 = rwsr-----

- Commandes : chown, chgrp, chmod, umask

Sécurité système – Protections des fichiers

Rappels des droits Unix sur les fichiers

- Les ACLs permettent sur un système de fichier POSIX de définir des listes d'accès non limitées au propriétaire, au groupe et au reste du monde
- Il est possible pour un fichier de rajouter un accès en lecture pour un utilisateur arbitraire
- Cette possibilité s'applique à tous les modes d'accès (rwx) sur tous les types de fichiers pour tous les utilisateurs et les groupes
- Commandes : setfacl, getfacl

- Tous les systèmes de fichiers ne supportent pas les ACLs, c'est le cas pour certaines versions de système de fichiers réseau (attention à NFS)
- Les outils de sauvegarde ne stockent pas les ACLs positionnées sur les fichiers (tar, cpio, rsync)

Sécurité système – chroot

Qu'est-ce que le chroot (change root) ?

- Il s'agit d'un appel système qui permet de modifier le point de départ de l'arborescence des fichiers, vu par un processus
- Cela permet de limiter (mettre en cage, jail sous BSD) l'environnement fichiers d'un processus, uniquement à ce qui lui est nécessaire
- On dissimule une partie du système de fichiers au processus et on ne lui laisse voir que ce dont il a besoin pour s'exécuter (exécutables, bibliothèques, fichiers de configuration, fichiers spéciaux, fichiers de logs)
- En pratique, on fait passer un répertoire du système pour la racine (*/*) de celui-ci, le processus ne pouvant pas (en théorie) accéder aux fichiers situés en dehors de son répertoire racine
- De cette manière, une intrusion réalisée à l'aide d'une vulnérabilité d'un service chrooté, serait contenu à son répertoire de chroot

Sécurité système – chroot

Qu'est-ce que le chroot (change root) ?

- Le chroot présente, cependant, un certain nombre d'inconvénients
  - Difficulté de réaliser un environnement (recherche des dépendances ou compilation en mode statique) et de le maintenir à jour
  - Cloisonnement au niveau du système de fichiers uniquement (aucune protection contre les vulnérabilités noyau)
  - Diverses méthodes pour sortir du chroot (accès ou création de fichiers de type bloc, accès à la mémoire, accès au réseau, …)

- Mais cette solution présente une première barrière surtout lorsqu'elle est proposée nativement et simplement par certains services (ssh)

Sécurité système – cgroups

- Les control groups (groupes de contrôle) correspondent à une fonctionnalité du noyau Linux permettant de
  - Limiter l'utilisation des ressources telles que le CPU, la mémoire, les entrées/sorties, le réseau, …
  - Isoler par la création d'espace de nommage (namespace) permettant de masquer les ressources (processus, connexions ou interface réseau, fichiers) entre espaces de nommage : un processus d'un espace de nommage A n'a aucune vision d'un processus d'un espace de nommage B ainsi que de ces ressources
  - Comptabiliser les ressources consommées par certains groupes (facturation)
  - Prioriser certains groupes pour leur accorder plus de ressources
  - Contrôler les groupes avec des notions d'arrêts, de points de sauvegardes et de redémarrage

- Les cgroups sont à la base de LXC (Linux Container) et de Docker

Sécurité système – su et sudo

- L'outil su (switch user) permet de changer d'identité (accès shell interactif) en passant par l'authentification du compte cible (utilisation des PAM)
- Par défaut, le compte cible est root
- C'est un outil qu'il est important de surveiller car il démontre
  - La volonté pour un utilisateur de changer d'identité
  - La connaissance des authentifiants d'un utilisateur B par un utilisateur A

- L'outil sudo permet d'autoriser l'exécution de commandes (en incluant le shell interactif) en changeant d'utilisateur avec ou sans authentification du compte source
- La politique d'autorisation de sudo est définie dans /etc/sudoers

```
frederic        ALL = (ALL) ALL
%wheel          ALL = NOPASSWD ALL
%mount          ALL = NOPASSWD /bin/mount, /bin/umount
```

Sécurité système – su et sudo

- Chaque utilisation de sudo est tracée (log) ce qui permet de limiter l'utilisation du compte générique root (qui offre peu de traçabilité)

- C'est aussi un outil à surveiller et pour lequel il ne faut pas donner trop d'accès

- Il est déconseillé d'utiliser le connecteur négatif (!) pour interdire l'utilisation d'une commande (principe de l'interdiction par défaut)

- Attention aux autorisations d'outils permettant le lancement de shell interactif ou aux autorisations de toutes les commandes présentes dans un répertoire (principe du système minimal et du strict besoin)

Sécurité système – Compte captif et shell restreint

- Afin d'appliquer le principe de séparation de privilèges, des utilisateurs de services (daemon) sont créés pour exécuter les processus de ces services (sshd, mysql, apache, …) et qui ne peuvent pas se connecter (shell à /bin/false)
- Il existe une autre catégorie d'utilisateurs qui ne dispose pas de shell interactif, ce sont les comptes captifs
- Le shell des comptes captifs est remplacé par une application qui lorsqu'elle s'arrêtera, déconnectera l'utilisateur
- Cette fonctionnalité permet de cloisonner un utilisateur à son strict besoin : accès à une application de soumission de jobs, de publication de contenu web, d'accès à une base de données, outil de redirection de flux réseau
- Attention à ce que l'application définie comme shell ne puisse pas donner un accès plus large au système (exécution de commandes annexes ou de shell interactif)

Sécurité système – Compte captif et shell restreint

- Un shell restreint est un shell interactif dont on limite les fonctionnalités afin de réduire l'accès au système
  - Limiter l'accès au répertoire HOME
  - Variables d'environnement non modifiables comme la variable PATH et fixées par un fichier de configuration
  - Exécution impossible de commandes en dehors du PATH
  - Limiter les redirection d'entrées/sorties

- Le shell rbash est la version restreinte de bash

Sécurité système – Chiffrement de système de fichiers

- Linux offre en natif un outil de chiffrement des systèmes de fichiers : LUKS (Linux Unified Key Setup) implémenté par cryptsetup
- Le chiffrement s'applique sur un raw device avant la création du système de fichiers
- Le mode de chiffrement utilisé est le XTS (qui ne doit pas être cyclique) avec AES
- La création d'un volume LUKS génère une clé de chiffrement aléatoire (de longueur cohérente avec le mode de chiffrement) qui est chiffrée avec la clé utilisateur
- Plusieurs clés utilisateurs peuvent être utilisées (la clé de chiffrement du volume est chiffrée avec chaque clé utilisateur)
- A chaque ouverture du volume chiffrée, une clé utilisateur est nécessaire pour déchiffrer le volume
- Le chiffrement est un bon mode de cloisonnement uniquement si le montage du volume intervient seulement au moment du besoin
- Problème récurrent du chiffrement : gestion de la clé (composition, brute force, stockage, séquestre, transmission, ...)

Sécurité système – DAC versus MAC

Discretionary Access Control

- Le propriétaire d'un objet spécifie quels sujets peuvent accéder à l'objet, c'est à la discrétion du propriétaire
- La plupart des système d'exploitation sont de type DAC
- Dans ce mode, c'est le propriétaire du fichier qui décide des droits d'accès du fichier et le système ne fait qu'appliquer ces droits

Madatory Access Control

- C'est le système qui spécifie quels sujets peuvent accéder à un objet donné, et non plus le propriétaire de l'objet
- Tout est basé sur des labels de sécurité (processus, ressources) et une politique qui définit qui peut accéder à quoi, au niveau système
- Il peut y avoir une relation d'ordre définie au niveau des labels afin d'implémenter différents niveaux de confidentialité
- SELinux, intégré au noyau Linux, implémente un MAC

Sécurité système – SELinux

- Security-Enhanced Linux permet le confinement d'applications (au niveau noyau) grâce aux labels de sécurité et la fin de la suprématie des accès root

- Labels de sécurité pour les sujets (processus, utilisateurs) et les objets (ressources)
  - utilisateur:role:type:niveaux

system_u:object_r:httpd_sys_content_t:s0

- Politique de sécurité pour définir qui a accès à quoi
  - allow|deny subject_type_t object_type_t:ressources { access}

allow httpd_t httpd_sys_content_t:file { read write create getattr setattr lock append unlink link rename open } ;

- Avec des règles de transition

domain_auto_trans(sysadmin_t, httpd_exec_t, httpd_t) ;

Sécurité système – SELinux

- Lorsqu'un accès (au niveau du noyau) est demandé, le moteur de décision de SELinux essaie de trouver une règle de la politique qui autorise (interdit) le sujet sur l'objet
  - si aucune règle n'est trouvée, l'accès est refusé
  - si une règle est trouvée, les protections fichiers standards sont vérifiées

- SELinux est très efficace pour le cloisonnement mais difficile à mettre en œuvre (réalisation et maintenance des politiques)

La sécurité informatique
Spécificités du HPC pour la sécurité
Grands principes de sécurité
Sécurité système

Sécurité réseau

Pare-feu, proxy, VPN, gestion des services

Surveillance du système
Exemples de menaces et d'attaques

Sécurité réseau – Pare-feu

- Un pare-feu se positionne en coupure d'une liaison réseau pour autoriser ou interdire la connexion
- Un pare-feu peut se positionner à plusieurs niveaux des couches (OSI) réseau, en général aux niveaux IP, TCP et UDP

- Un pare-feu peut être à sans état (stateless) ou à états (statefull) afin de suivre les états de la couche TCP
- Un pare-feu peut tracer certaines couches applicatives pour autoriser les flux négociés par certains protocoles (ftp passif)

- Un pare-feu peut gérer le NAT (Network Address Translation) en sortie comme en entrée
- Les adresses IP sont échangés entre l'entrée et la sortie (et inversement) pour différentes raisons
  - masquer le réseau interne
  - utiliser des adresses non routables en interne
  - rediriger les flux réseau vers des machines dédiées

Sécurité réseau – Pare-feu

- Le pare-feu étant un SPF (Single Point of Failure), une configuration en haute disponibilité est souvent nécessaire (avec synchronisation des états)

- Les logs générés par les pare-feux sont toujours très intéressants et doivent être collectés pour analyses

- Ne pas confondre un pare-feu et un IDS ou un IPS (Intrusion Detection System ou Intrusion Prevention System)

- Les pare-feux peuvent être sujet aux DOS ou DDOS (Denial Of Service ou Distributed Denial Of Service)

- Linux intègre un pare-feu nativement dans le noyau
  - ipfilter avec sa commande de configuration iptables
- ipfilter peut être utilisé comme pare-feu local à la machine mais aussi comme pare-feu standard avec transfert des paquets entre interfaces réseau

Sécurité réseau – Proxy

- Un proxy, à la différence d'un pare-feu, fonctionne au niveau de la couche applicative du réseau
- Un proxy permet de vérifier qu'un protocole applicatif est bien respecté tout en autorisant ou interdisant certaines requêtes
- Exemple d'un proxy HTTP : squid
  - Vérifie le protocole HTTP (et HTTPS ?)
  - Permet de mettre des restrictions sur les domaines accédés, sur les méthodes utilisées ou sur les URLs
  - Permet l'authentification des utilisateurs (traçabilité et autorisation)
  - Est redondant avec un pare-feu (sécurité en profondeur)
- Exemple d'un proxy TCP ou UDP : socks
  - Vérifie le protocole TCP ou UDP
  - Permet l'authentification des utilisateurs (traçabilité et autorisation)
  - Est redondant avec un pare-feu (sécurité en profondeur)
- Certains protocoles n'ont pas de besoin de proxy du fait de leur aspect réparti : DNS, SMTP, NTP

Sécurité réseau – VPN

- Un Virtual Private Network (Réseau Privé Virtuel) permet de relier un réseau à un autre réseau comme s'il s'agissait du même réseau local
- Différents types de VPN existent :
  - Réseau à réseau
  - Machine à réseau
  - Machine à machine
- Un VPN est basé sur l'encapsulation réseau permettant de transporter un paquet d'un réseau local vers un autre réseau local
- Un VPN peut être chiffré et authentifié, mais ces propriétés ne sont intrinsèques à un VPN
- Un VPN chiffré (et authentifié) permet de relier deux réseaux en passant par une infrastructure réseau qui n'est pas de confiance
- Un VPN peut être réalisé à différentes couches du modèle OSI
  - PPP, PPPoE, PPTP, L2TP, GRE, IPSec, SSL/TLS, SSH
- Afin de gérer plusieurs niveaux d'authentification, il est possible d'imbriquer des VPN : L2TP/IPSec

Sécurité réseau – Gestion des services

- Les services (daemon) réseau d'un serveur doivent respecter le principe du système minimal et d'une utilisation au strict besoin
- Les services réseau n'ont pas forcément besoin d'écouter sur toutes les interfaces du serveur, d'être accédés depuis partout ou de fonctionner avec les privilèges root

- Un service réseau doit être correctement configuré
  - strict besoin, séparation de privilèges
  - cloisonnement système, protection par un pare-feu local ou les tcpwrappers
  - connexion authentifiée (TLS, SSH) et correctement instrumenté (logs)

- L'utilisation de xinetd pour gérer les services ne répondant pas à des exigences de sécurité est à préconiser
  - double-reverse DNS lookup, ACLs réseau (tcpwrappers)
  - logs des événements, heures d'accès au service, choix de l'interface en écoute, limite de connexions et des ressources système, ...

La sécurité informatique
Spécificités du HPC pour la sécurité
Grands principes de sécurité
Sécurité système
Sécurité réseau

Surveillance du système

Syslogd, auditd, intégrité des logs, accouting et collectd, audits et test de pénétration

Exemples de menaces et d'attaques

Surveillance du système - syslogd

- Le service syslog (syslogd, syslogd-ng, rsyslogd) permet de collecter les événements du système (userland et kernel via klogd) et de les ventiler dans des fichiers de logs
- Le service syslog gère les logs locaux, mais aussi distants (port UDP ou TCP 514)
- Tous les éléments d'un SI génèrent des logs (serveurs, stations, pare-feu, proxys, portails VPN, routeurs, …)

- Les messages sont étiquetés
  - par famille (facility) : auth, daemon, kern, mail, …
  - par priorité (priority) : info, warn, err, notice, …
- Chaque message de log est daté et contient l'hôte qui l'a généré

- Les versions les plus récentes de syslog (rsyslog) permettent de ventiler les événements selon le couple facility/priority, l'hôte et d'autres éléments
- Le protocole TCP est à préférer (à UDP) pour le transport des logs, d'un point de vue sécurité

Surveillance du système - syslogd

- Les logs collectés doivent être analysés régulièrement en complément d'une analyse automatique générant des alertes et/ou des rapports quotidien
- L'utilisation d'outils d'archivage, de tri, de recherche, de visualisation et de corrélation est à préférer (logrotate, logsurfer, logcheck, logwatch, …)

- Le volume de données à traiter peut être gigantesque et l'utilisation de technologie de data mining pour le stockage et la corrélation peut être nécessaire

- Il faut, également, penser à récupérer les logs de SELinux

- Enfin, ce qui est fait pour les logs systèmes, peut être étendu aussi au réseau avec une capture des flux réseau aux endroits stratégiques pour analyses (détections, alertes, rapports)

Surveillance du système - auditd

- Le service noyau auditd de Linux permet de générer des logs suite aux appels système effectués
- La configuration consiste à définir des filtres permettant de créer un log sur certains critères (nom de l'appel système, chemin de l'exécutable, nom de fichier accédé, uid, gid, règle iptables, …)
- Le log généré est envoyé au processus audisp pour traitement (action, envoi de logs à syslogd, …)
- Le service auditd est un service de surveillance, pas de contrôle

Surveillance du système - auditd

- Exemples de logs intéressants
  - execve (commandes lancées avec les paramètres par utilisateur)
  - iptables (connexions réseau par utilisateur)
  - fichiers (fichiers accédés en écriture dans des répertoires particuliers)
  - avc (accès refusés par SELinux)

- Inconvénients d'auditd
  - Difficile à configurer (bonnes connaissances système nécessaires)
  - Volume de logs important
  - Impact sur les performances non négligeable, les filtres doivent être adaptés à la surveillance et optimisés

Surveillance du système – Intégrité des logs

- Les logs sont importants pour la surveillance et leur intégrité doit être préservée
- Il est, donc, important de réfléchir à des moyens de limiter l'interception ou la modification de ces événements
- Une architecture dédiée doit être mise en place pour la gestion des logs (chemin sécurisé d'acheminement des logs)
- Ne pas oublier que le protocole syslog est un protocole nativement réparti et qu'un log peut passer par plusieurs serveurs avant d'atteindre sa destination
- Des solutions de chiffrement et/ou d'authentification (TCP) peuvent être utilisées mais attention aux performances (risque d'écrouler les serveurs et de perdre des logs)
- L'utilisation de techniques de heartbeat pour détecter les problèmes d'acheminement de logs peuvent être utilisés
- Les solutions de diodes réseau physiques permettent de sécuriser le stockage et l'analyse des logs (connexion réseau unidirectionnelle) mais avec un coût non négligeable (port monitoring moins cher mais moins fiable)

Surveillance du système – Accounting et collectd

- L'accounting (comptabilité) permet de récupérer l'utilisation des ressources (temps CPU, entrées/sorties, occupation disques) par utilisateur, par groupe, par population afin de générer la facturation
- L'accounting est activé au niveau du noyau pour l'hôte local, dans un environnement HPC, l'accounting doit être consolidé au niveau du calculateur et/ou du centre de calcul
- Les données brutes d'accounting peuvent être très intéressantes (commandes lancées par les utilisateurs avec l'occupation des ressources) et sont à intégrer à la surveillance

- Le service collectd (client) récupère l'utilisation des ressources (CPU, mémoire, entrées/sorties disques, réseau) pour toute la machine
- Et les envoie au serveur collectd qui stocke les informations pour cette machine
- Plusieurs outils sont disponibles pour visualiser les données stockées par collectd (via un navigateur par exemple)

Surveillance du système – Audits

- Un audit est un ensemble de procédures visant à vérifier un état vis-à-vis d'un référentiel à un instant t
- En sécurité informatique, un audit consiste à mesurer la divergence entre la politique de sécurité définie et les mesures techniques et organisationnelles à mettre en place et la réalité
- Le but final d'un audit est de pouvoir corriger la divergence mesurée pour la minimiser et se rapprocher du référentiel défini
- Un audit est un processus principalement organisationnel qui peut avoir une composante technique plus ou moins faible

- Un audit de vulnérabilités est beaucoup plus technique et vise à recenser les vulnérabilités présentes et/ou exploitables dans un système
- Une façon de réaliser cette audit de vulnérabilités est de confronter les versions des logiciels installés avec les mises à jour disponibles de ces même logiciels (=> mises à jour)
- Des outils permettant cette comparaison sont disponibles et même fournis par les éditeurs de distribution

Surveillance du système – Test de pénétration

- Un test de pénétration est un exercice visant à tester la résistance d'un système à une attaque à un instant t

- Les testeurs prennent le rôle d'attaquant selon un périmètre pré-défini (interne, externe, en boite blanche, en boite noire, …)

- Ils attaquent le système en essayant d'acquérir le plus de privilèges possible, de données possibles ou de rebondir vers le plus de cibles selon ce que l'on souhaite tester

- A l'issue du test de pénétration, un rapport est rédigé intégrant des préconisations afin de supprimer ou de limiter les vulnérabilités utilisées (techniques et/ou organisationnelles)

La sécurité informatique
Spécificités du HPC pour la sécurité
Grands principes de sécurité
Sécurité système
Sécurité réseau
Surveillance du système

Exemples de menaces et d'attaques

Usurpation d'identité, ip spoofing, buffer overflow, ingénierie sociale et phishing, remote exploit, rootkit, race condition, ransonware

Exemple de menaces et d'attaques

Usurpation d'identité

IP spoofing

Buffer overflow

Ingénierie sociale

Local exploit

Rootkit

Race condition

SQL injection

Cross-site scripting

Man-in-the-Middle

Arp spoofing

Stack / Heap overflow

Phishing

Remote exploit

Ransonware

Brute force

DE LA RECHERCHE À L'INDUSTRIE

cea

Questions ?

ensiie
Paris-Évry | Strasbourg

Commissariat à l'énergie atomique et aux énergies alternatives
Centre de Saclay | 91191 Gif-sur-Yvette Cedex
T. +33 (0)1 69 26 40 00 | F. +33 (0)1 69 26 XX XX

DAM
DSSI
SISR

Etablissement public à caractère industriel et commercial | RCS Paris B 775 685 019