Approches développées au LAAS :

Tolérance aux intrusions
Évaluation quantitative de la sécurité

Yves Deswarte
LAAS-CNRS, Toulouse, France

Approches développées au LAAS :

Tolérance aux intrusions
Évaluation quantitative de la sécurité

Yves Deswarte & David Powell
LAAS-CNRS, Toulouse, France

Utilisateurs Internet

Utilisations :
B2B, B2C, C2A, e-government, associations, communautés virtuelles, usage privé…

Buts :
commerce, administration, démocratie, société, culture, loisirs, …

On ne doit pas exclure une catégorie d'utilisateurs au bénéfice d'une autre

Différents besoins ⇒ différents niveaux de sécurité

État de fait

1. Il y a des machines mal administrées qui peuvent être exploitées par des attaquants pour accroître leurs capacités et cacher leurs traces
2. Il y a des centaines de millions d'utilisateurs d'Internet dont une très petite proportion sont des attaquants potentiels

Techniques classiques de la sécurité

Authentification

- Identifier les utilisateurs
- Les tenir responsables (audit)

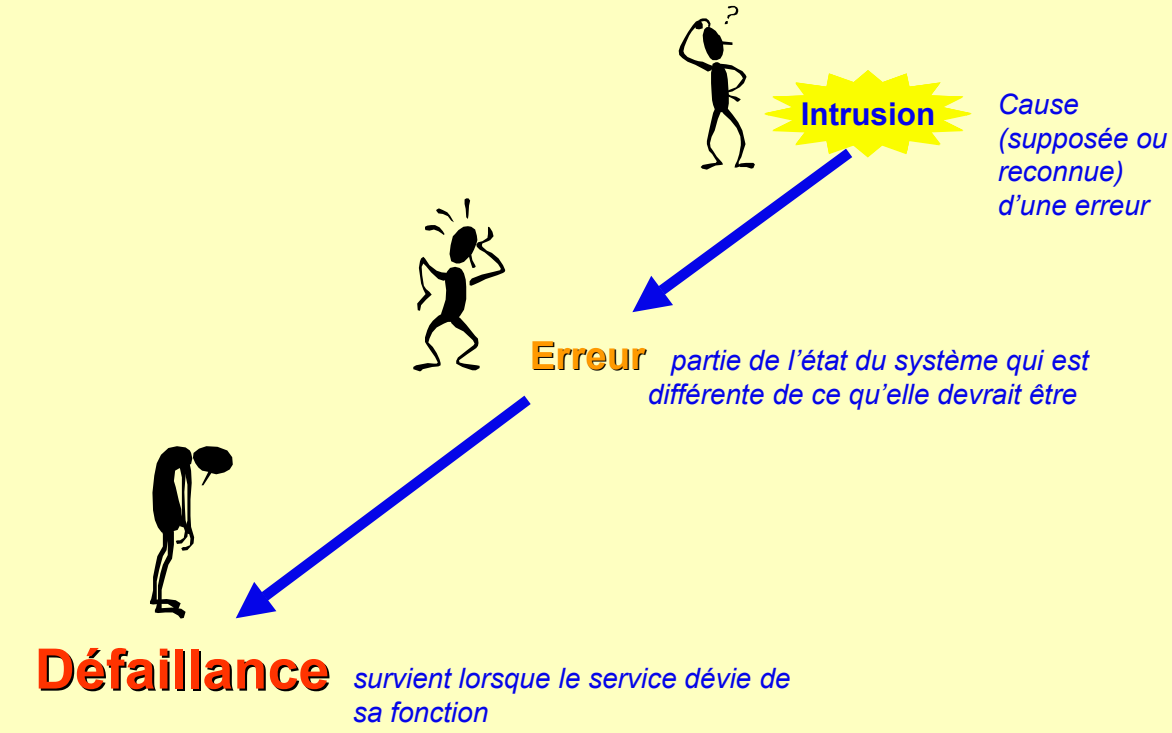
Autorisation

- Empêcher les actions illégitimes
- Principe du moindre privilège
  légitime ⇔ nécessaire

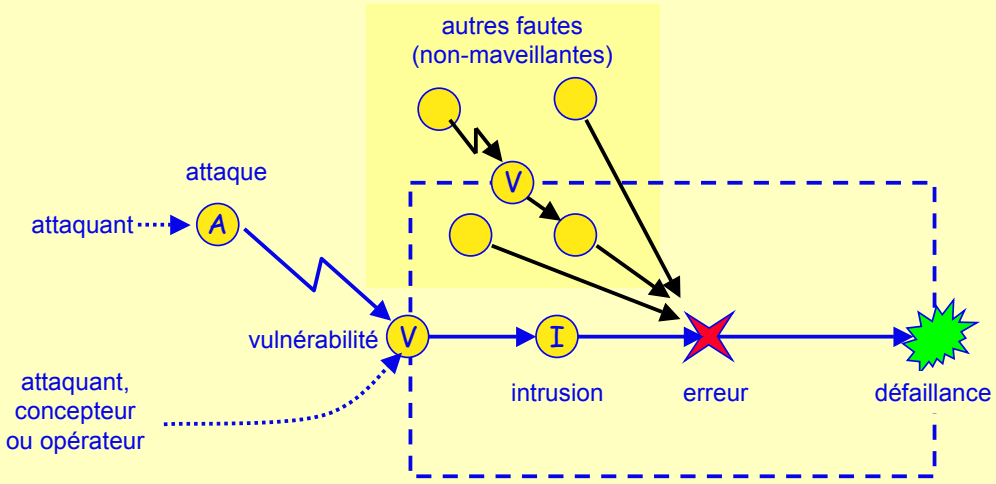
Dissuasion ⇐ Punition ⇐ Détection

➢ **Inefficace dans le contexte Internet :**
- o Authentification forte infaisable pour les sites publics
- o Applications et OS sur étagères :
  - nombreuses failles
  - rustines non appliquées : manque de temps ou de compétence, crainte de perdre des fonctionnalités nécessaires
- o Les protocoles Internet sont vulnérables (héritage Arpanet)
- o La pression économique ne favorise pas (encore) des défenses connues :
  - filtrage d'entrée (ingress filtering), capacité de traçage, …

Concepts de base de la SdF



Modèle de faute



- **attaque** - activité malveillante externe visant à violer une ou plusieurs propriétés de sécurité; une tentative d'*intrusion*
- **vulnérabilité** - faute (par malveillance ou maladresse), dans les spécifications, la conception, la réalisation, l'installation ou la configuration du système, ou dans la façon de l'utiliser, qui peut être exploitée pour produire une *intrusion*
- **intrusion** - faute (malveillante) résultant d'une *attaque* qui a réussi à exploiter une *vulnérabilité*

Méthodes de sûreté de fonctionnement

FOURNITURE

Prévention des fautes - comment empêcher que des *fautes* surviennent ou soient introduites

Tolérance aux fautes - comment fournir un service conforme à la fonction en dépit des *fautes*

VALIDATION

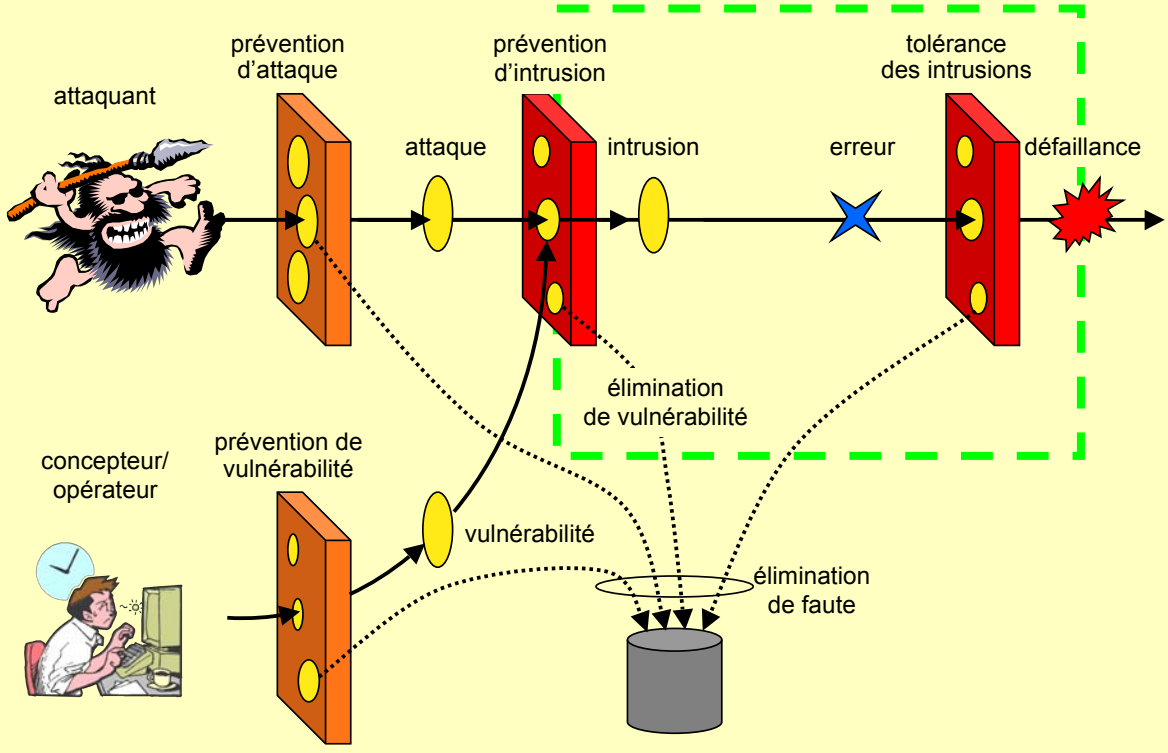
Élimination des fautes - comment réduire la présence (en nombre ou en gravité) des *fautes*

Prévision des fautes - comment estimer la présence, la création et les conséquences des *fautes*

Éviter les fautes

Accepter les fautes

Prévention, tolérance et élimination



Tolérance aux intrusions

- Les intrusions sont des fautes
- Les fautes peuvent être tolérées

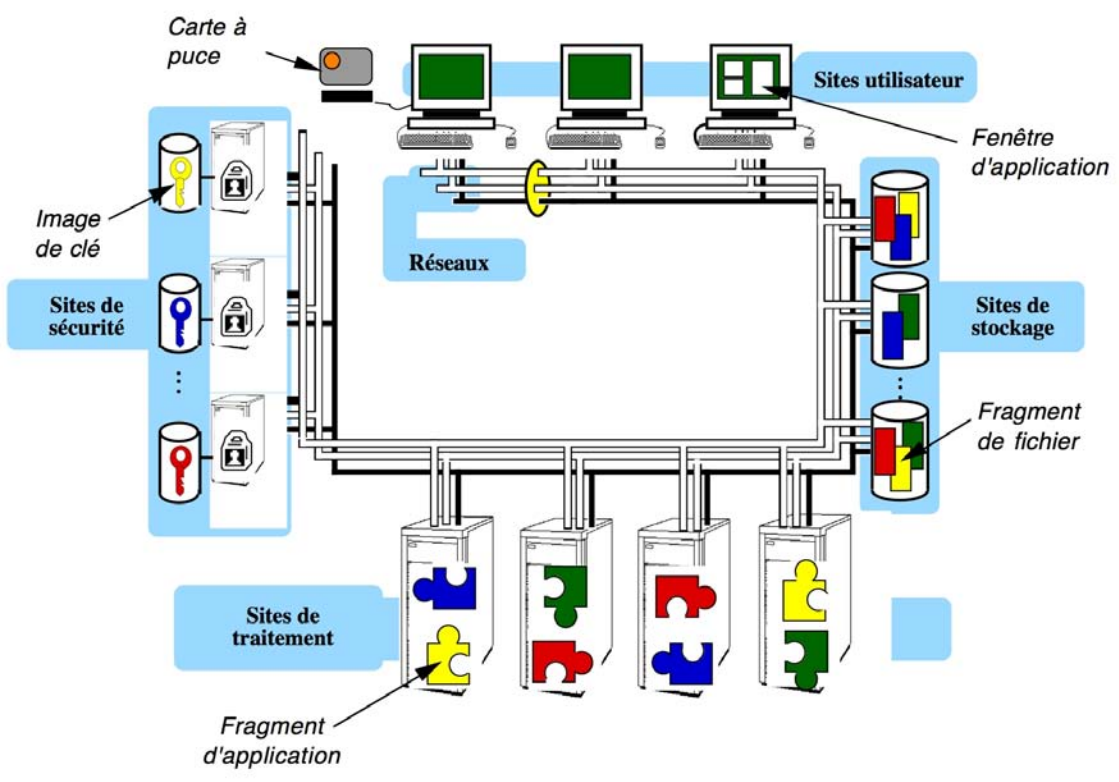
- Mais:
  - on ne peut pas se reposer sur la faible probabilité d'attaques presque simultanées sur différentes parties du système !
- Il faut donc s'assurer que :
  - chaque partie est suffisamment protégée (pas d'attaque triviale)
  - l'intrusion d'une partie ne facilite pas l'intrusion d'autres parties
    - ↪ une intrusion ne doit pas révéler des données confidentielles

Masquage d'intrusion

- Une intrusion dans une partie du système ne doit donner accès qu'à des informations non-significatives

- FRD : Fragmentation-Redondance-Dissémination
  - **Fragmentation**: fragmenter l'information de telle sorte qu'un fragment isolé ne contienne pas d'info sensible : **confidentialité**
  - **Redondance** : ajouter de la redondance de telle sorte que la modification ou destruction de fragments n'empêche pas les accès légitimes : **intégrité** + **disponibilité**
  - **Dissémination** : isoler les fragments individuels
    - topologie/fréquences
    - temps
    - privilèges

Fragmentation-Redondance-Dissémination



Projet MAFTIA

FP5 IST Dependability Initiative
Cross Program Action
Dependability in services and technologies

Malicious- and Accidental-Fault Tolerance for Internet Applications

University of Newcastle (UK)	Brian Randell, Robert Stroud
University of Lisbon (P)	Paulo Verissimo
DSTL + QinetiQ (ex-DERA) (UK)	Tom McCutcheon, Sadie Creese
University of Saarland (D)	Birgit Pfitzmann
LAAS-CNRS, Toulouse (F)	Yves Deswarte, David Powell
IBM Research, Zurich (CH)	Marc Dacier, Michael Waidner

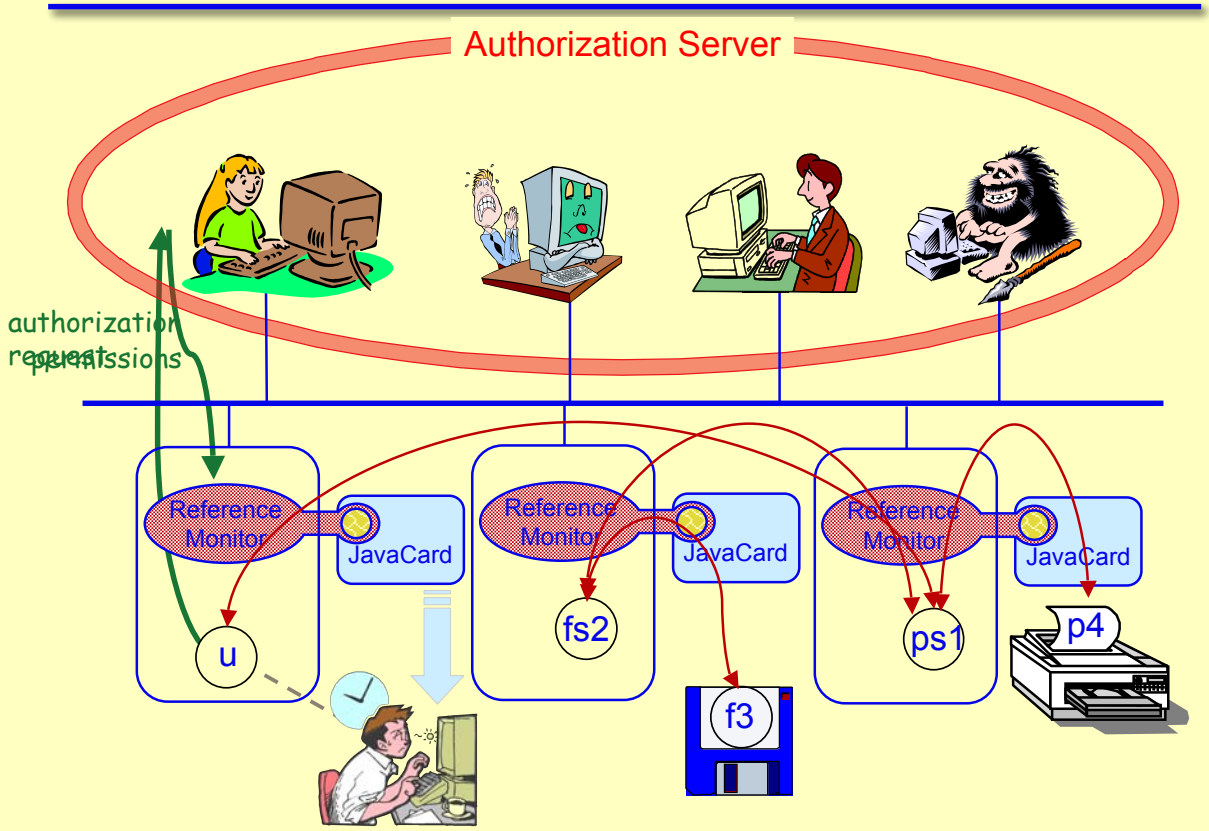
~ 55 personnes-ans, finacement CE ~2.5M€
Jan. 2000 -> Déc. 2002 (Fév. 2003)

Résultats obtenus par MAFTIA

- Cadre conceptuel et modèle d'architecture
- Mécanismes et protocoles :
  - **Intergiciel (*middleware*) pour la tolérance aux intrusions**
    - communications multicast (ordre causal, temps-réel) sécurisées
  - **Système de détection d'intrusion à large échelle**
  - **Tierces parties de confiance tolérant les intrusions**
  - **Mécanismes d'autorisation distribuée (TAI)**
- Validation et évaluation

http://www.maftia.org/

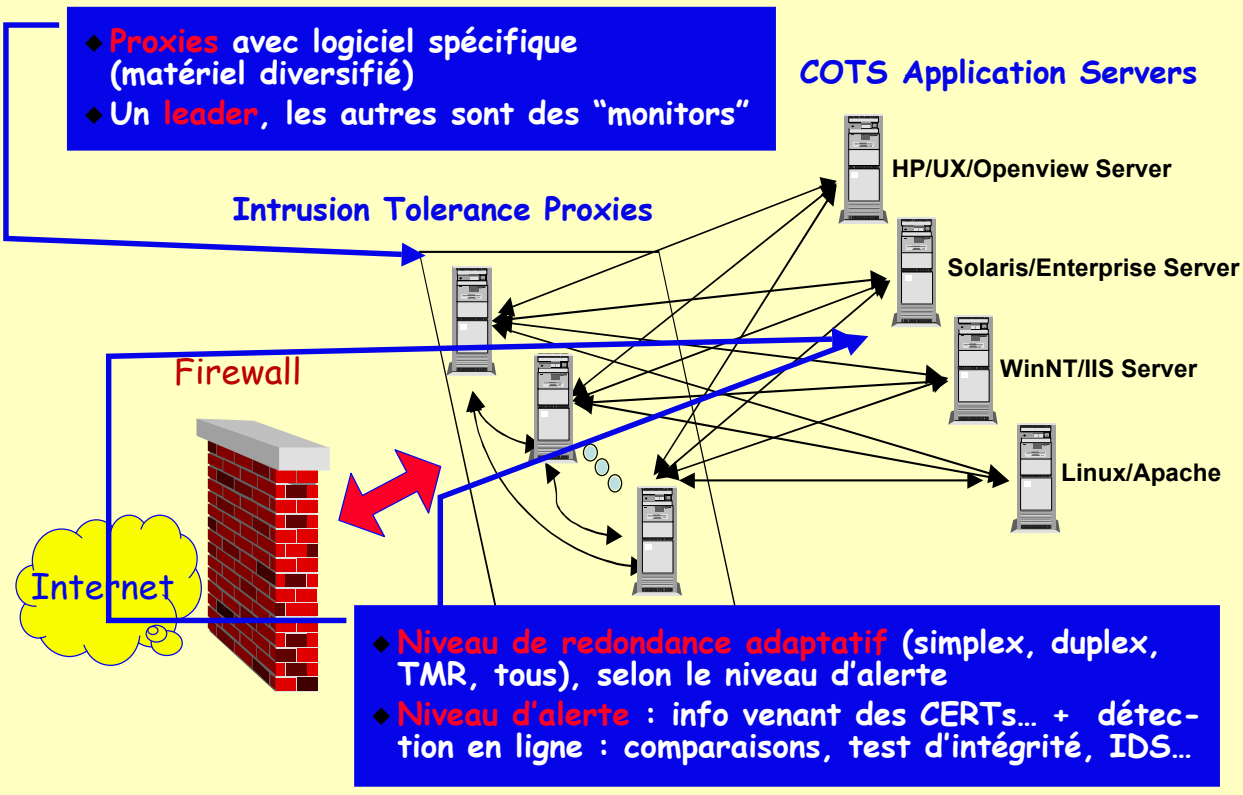
Schéma d 'autorisation MAFTIA



Projet DIT

- DIT = Dependable Intrusion Tolerance
- Programme DARPA OASIS (Organically Assured and Survivable Information Systems)
- En partie sous-contracté au LAAS par SRI-International
- Conception et réalisation d'un serveur Web tolérant les intrusions

Architecture DIT



Conclusion

- Étant donnés
  - le taux d'attaques actuel sur Internet
  - le grand nombre de vulnérabilités des systèmes courants
- la tolérance aux intrusions est une technique prometteuse
  - compatible avec les systèmes classiques (COTS)
  - avec une redondance matérielle modérée et peu de logiciel spécifique
- mais coûteuse
  - support de plateformes multiples et diversifiées
    ⇒ indépendance de vulnérabilité
  - opérateurs/administrateurs indépendants
    ⇒ tolérance des attaques internes
- Est-ce le prix de la sécurité dans un monde ouvert et incertain?

Approches développées au LAAS :

Tolérance aux intrusions
Évaluation quantitative de la sécurité

Yves Deswarte
LAAS-CNRS, Toulouse, France

Approches développées au LAAS :

Tolérance aux intrusions

Évaluation quantitative de la sécurité

Méthodes classiques d'évaluation - 1

- **Analyse de risque**
  - **Identifier les vulnérabilités**
  - **Identifier les menaces**
  - **Identifier les attaques possibles**
    - Imaginer des scénarios permettant d'exploiter les vulnérabilités
    - Estimer leur fréquence, en fonction des capacités de l'attaquant
    - Estimer les dommages correspondants
  - **Calculer les espérances de pertes (fréquence x dommages)**
  - **Identifier des contre-mesures et leur coût**

- **Méthodes analytiques (MARION, MELISA, MEHARI,…), mais paramètres estimés par les experts**

- **Processus long et coûteux**

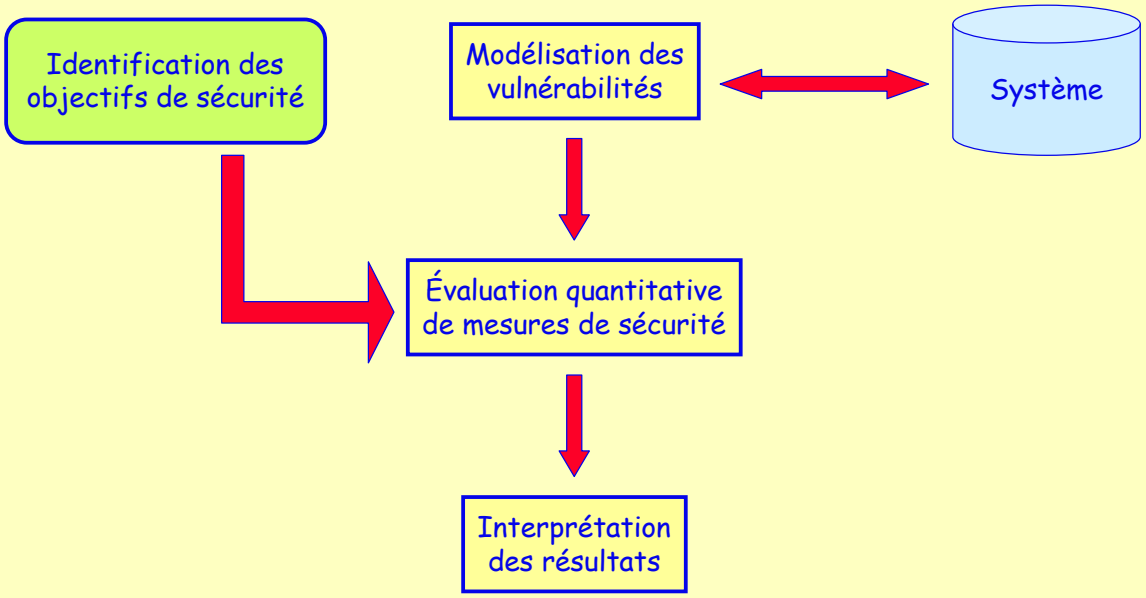
Méthodes classiques d'évaluation - 2

- Critères d'évaluation (TCSEC, ITSEC, CC)
  - Fonctionnalités de sécurité
    - Définition des besoins
    - Caractérisation des mécanismes répondant à ces besoins
    - Estimation de la force des mécanismes
  - Niveaux d'assurance (confiance)
    - Exigences sur les méthodes de développement et de vérification
    - Documentation
  - Profils de protection
- Qualitatif plutôt que quantitatif
- Vision plutôt statique: comment le système a été conçu, plutôt que comment il est utilisé

Mesures sur le système en opération

- Objectifs :
  - surveiller les évolutions de la sécurité en fonction des modifications de configuration et d'usage
  - identifier les meilleures améliorations possibles
  - négocier avec les utilisateurs
- Mesure = caractéristique du système
  - qu'il y ait ou non une attaque, la mesure est la même
- Robustesse du système face à toutes les attaques possibles
  - "effort moyen" que doit déployer un attaquant pour mettre en défaut les objectifs de sécurité
  - effort = variable pluri-dimensionnelle : temps, compétence (connaissance, intelligence), équipement nécessaire, …

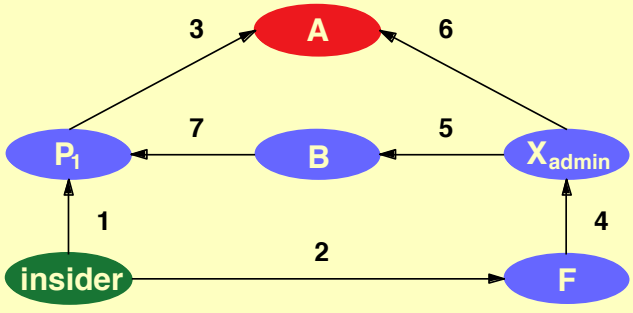
Approche



Modélisation des vulnérabilités

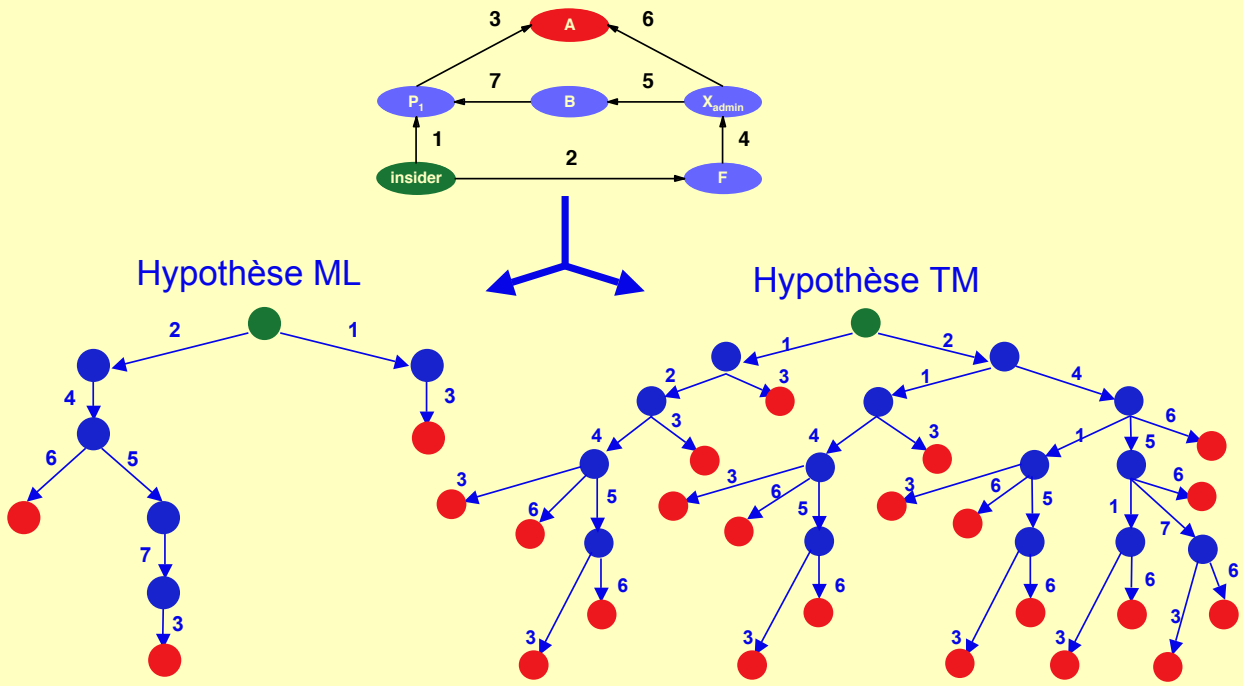
❖ Graphe des privilèges

- Nœud = ensemble de droits (d'un utilisateur, d'un groupe d'utilisateurs…)
- Arc = méthode pour acquérir des privilèges (faille ou utile)
- Poids/arc = effort pour exploiter la méthode



Pour un arc X→Y :
1) X peut deviner le mot de passe d'Y
2) X peut installer un cheval de Troie pour Y
3) X peut exploiter une faille du mailer d'Y
4) Y est un sous-ensemble d'X
5) Y utilise un programme qu'X peut modifier
6) X peut modifier un programme "s-uid" d'Y
7) X est dans le `.rhosts` d'Y

Recherche des scénarios



Différentes mesures

- METF-ML :
  Effort moyen selon l'hypothèse ML
- METF-TM :
  Effort moyen selon l'hypothèse TM
- NP : Nombre de chemins
- SP : Plus court chemin

ÉSOPE (Évaluation de la sécurité opérationnelle)

- Ensemble d'outils logiciels pour :
  - Définir les objectifs de sécurité (outil graphique)
  - Identifier les vulnérabilités automatiquement (réseau de machines Unix)
  - Calculer les mesures
  - Aider à exploiter les résultats

Expérimentation : objectifs

- Validation de l'approche :
  - évaluer la pertinence des mesures en fonction de l'évolution de la sécurité
  - étudier la faisabilité sur un système réel
- *Ne faisait pas partie des objectifs* :
  - corriger les vulnérabilités identifiées

Contexte de l'expérimentation

Système-cible :

- Unix
- 700 utilisateurs - 200 machines - réseau local
- 13 mois (juin 1995 - juillet 1996)

13 types de vulnérabilités étudiés (fichiers `.rhosts`, `.*rc`, mots de passe, etc.)

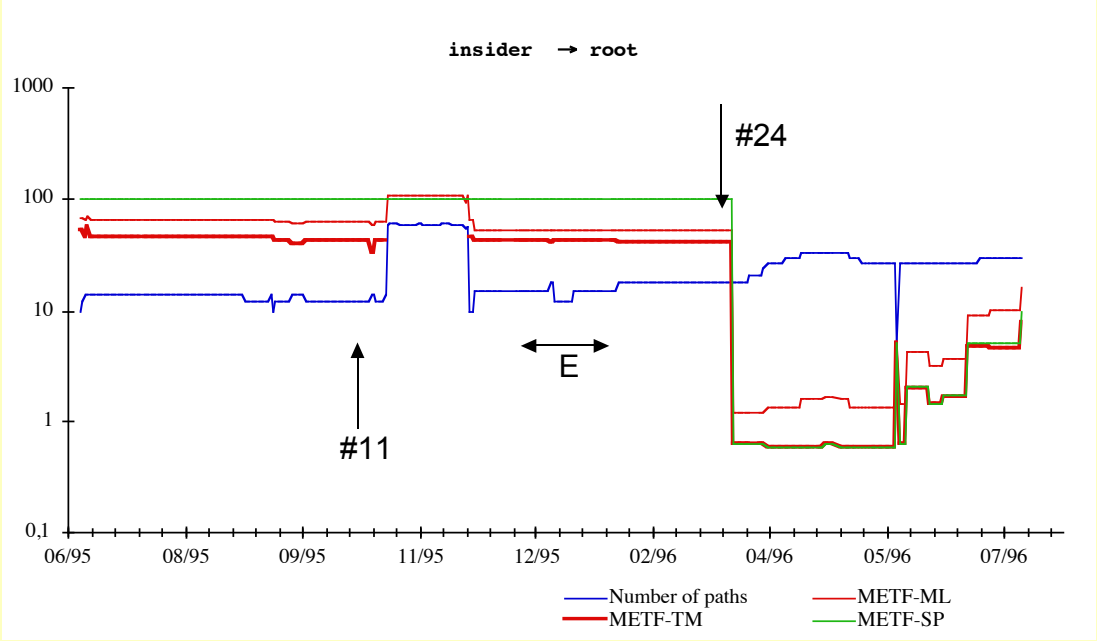
Objectifs:

	Attaquant	cible
objectif 1	`insider`	`root`
objectif 2	`insider`	`admin_group`

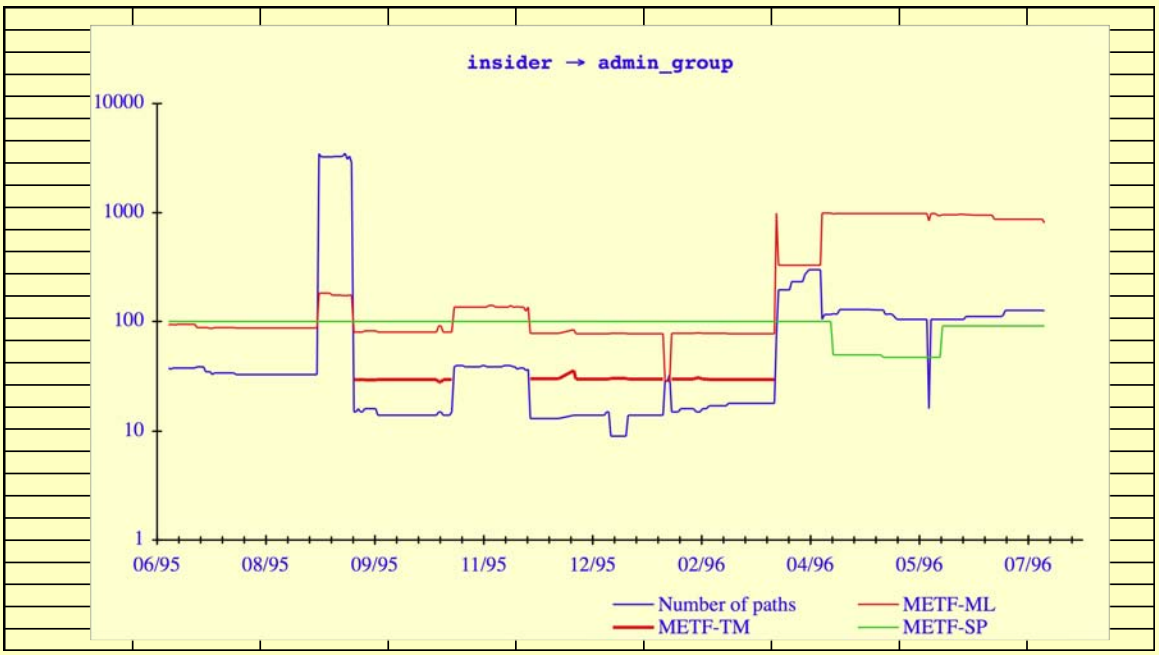
4 niveaux de difficulté :

Type	Poids
`immédiat`	10
`facile`	10^2
`difficile`	10^3
`très difficile`	10^4

Résultats expérimentaux (1)



Résultats expérimentaux (2)



Critique des mesures

- Le plus court chemin (SP) n'est pas assez sensible pour révéler des événements importants
- Le nombre de chemins (NP) change souvent et produirait un grand nombre de fausses alarmes.
- METF-ML présente une bonne sensibilité aux événements importants.
- METF-TM permet une interprétation plus facile, mais est parfois trop complexe à calculer.

Conclusion

- Une approche objective pour l'évaluation quantitative de la sécurité
- Des mesures qui ne sont pas absolues :
  - On ne peut comparer les mesures de systèmes différents
- Outil complémentaire des autres outils de sécurité
- Validation du modèle d'attaques : observations à l'aide de Honeypots (projet CADHo, ACI Sécurité Informatique)

Autres apports SdF <-> Sécurité

- Application de modèles *"security"* à la *"safety"* : co-existence de logiciels de criticités multiples (projet européen GUARDS)
  - Contrôle de flux (Biba) : haute intégrité → basse intégrité
  - Tolérance aux fautes : basse intégrité → haute intégrité
  - Noyau de sécurité
- Critères d'évaluation de la sûreté de fonctionnement projet européen SQUALE
  - Indépendants du domaine d'application, mais compatibles avec les normes spécifiques aux domaines (CC, DO178B, CEI 880, 61508…)
  - Couvrant les aspects safety, security, disponibilité, intégrité …
  - Prise en compte des COTS, de la tolérance aux fautes

< http://www.research.ec.org/squale/ >

λ/μ 98, SAFECOMP 99