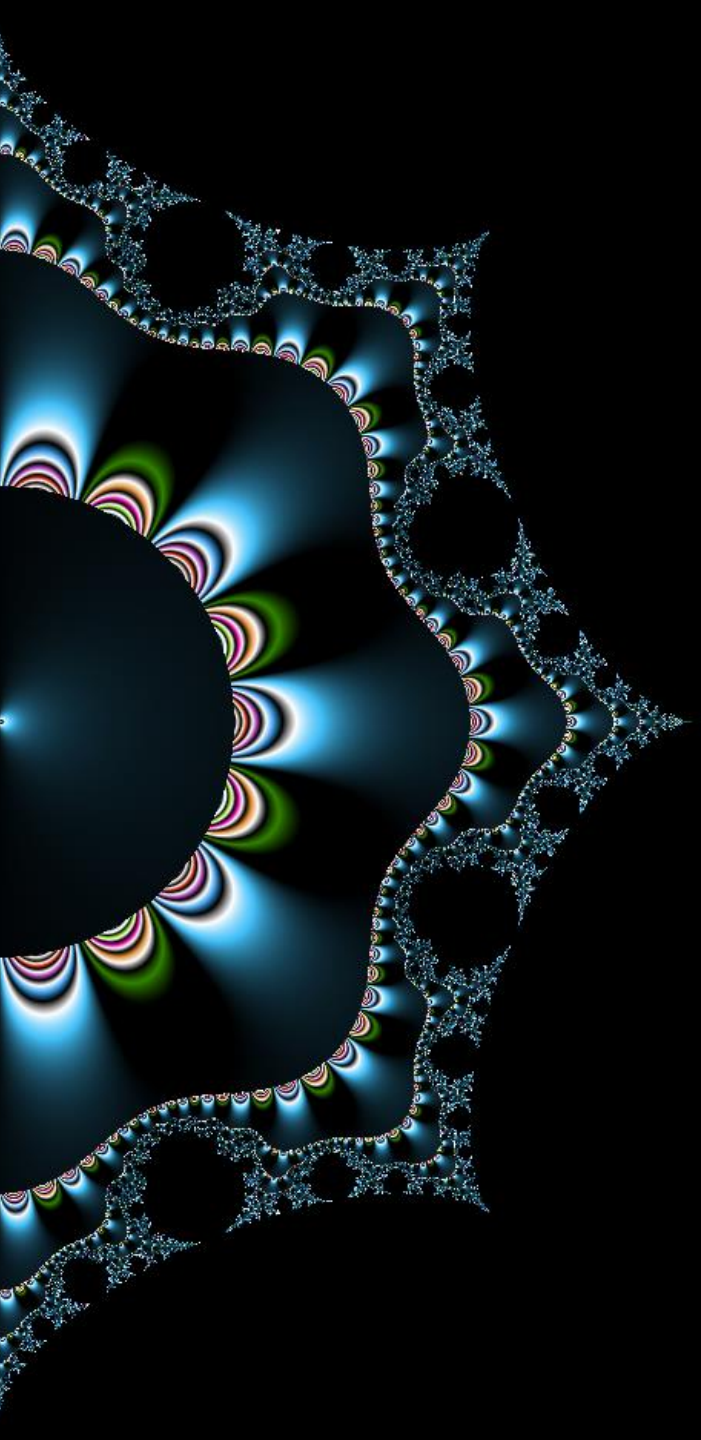


Authentification

Prouver qui on est

Thibaut HENIN

tbowan@arsouyes.org

Petite définition

Identification

« Qui on est »

Authentification

« Le prouver »

Facteurs d’authentification

Je sais

(Mot de passe)

Je suis

(Biométrie)

Je possède

(Téléphone, carte à puce, …)

Je sais faire

(Signature manuscrite)

Méthode d’authentification

Simple

(1 facteur)

Forte

(au moins 2)

Contextuelle

Adresse IP, HTTP Referer, HTTP Host, Cookies

N’est pas sûr

Adresse IP

```
<VirtualHost *:80>

  ServerName  secured.example.com

  <Location />
    Require ip 87.98.129.198
    Require ip 192.168.0.0/24
    Require host trusted.example.com
    Require forward-dns other.example.com
    Require local
  </Location>

</VirtualHost>
```

Géolocalisation

```
<VirtualHost *:80>

  ServerName  secured.example.com

  GeoIPEnable On
  GeoIPDBFile /path/to/GeoIP.dat

  <Location />
    Deny from GEOIP_COUNTRY_CODE=CN
    Deny from GEOIP_COUNTRY_CODE=RU
  </Location>

</VirtualHost>
```

Limitations sur les adresses

Man In the Middle

IP & DNS Spoofing

Proxy sortant

En-tête « x-Forwared-For »

Mise à jour des bases de données

En-tête HTTP : HOST

```
<VirtualHost *:80>

  ServerName  intranet.example.com

  Require expr "%{HTTP_HOST} == 'intranet.example.com'"

</VirtualHost>
```

En-tête HTTP : Referer

```
<VirtualHost *:80>

  ServerName  intranet.example.com

  Require expr
        "%{HTTP_REFERER} -strmatch '*://%{HTTP_HOST}/*'"

</VirtualHost>
```

En-tête HTTP : User Agent

```
<VirtualHost *:80>

  ServerName  intranet.example.com

  Require expr "! %{HTTP_USER_AGENT} -strmatch '*NESSUS*'"

</VirtualHost>
```

Cookies

```php
<?php

function setAsAdmin() {
    $_COOKIES['admin'] = true ;
}

function isAdmin() {
    return @$_COOKIES['admin'] === true ;
}
```

Limitations sur les en-têtes

Forgées par le client

Secret partagé

Mots de passes, OTP, cryptographie

Mot de passe

Principe

Stockage « identifiant / mot de passe »

Sécurité spécifique

Stockage & choix

Stockage du mot de passe

Défense en profondeur

(en cas de compromission de la base)

Assurer la confidentialité

(même si la base est lisible)

Stockage du mot de passe

Niveau 0	Attaque
En clair	Triviale

Stockage du mot de passe

Niveau 1

Hacher le mot de passe

`sha512(password)`

Attaque

Dictionnaires

(dont google)

Rainbow tables

(dictionnaire compressé)

Stockage du mot de passe

Niveau 2

Saler le mot de passe

`sha512(sel + password)`

Attaque

Brute force

(hashcat)

Stockage du mot de passe

Niveau 3

Hachage ralenti

`Bcrypt(sel + password)`

`Password_hash(password)`

Attaque

Brute force

(hashcat mais bien plus lent)

Choix du mot de passe

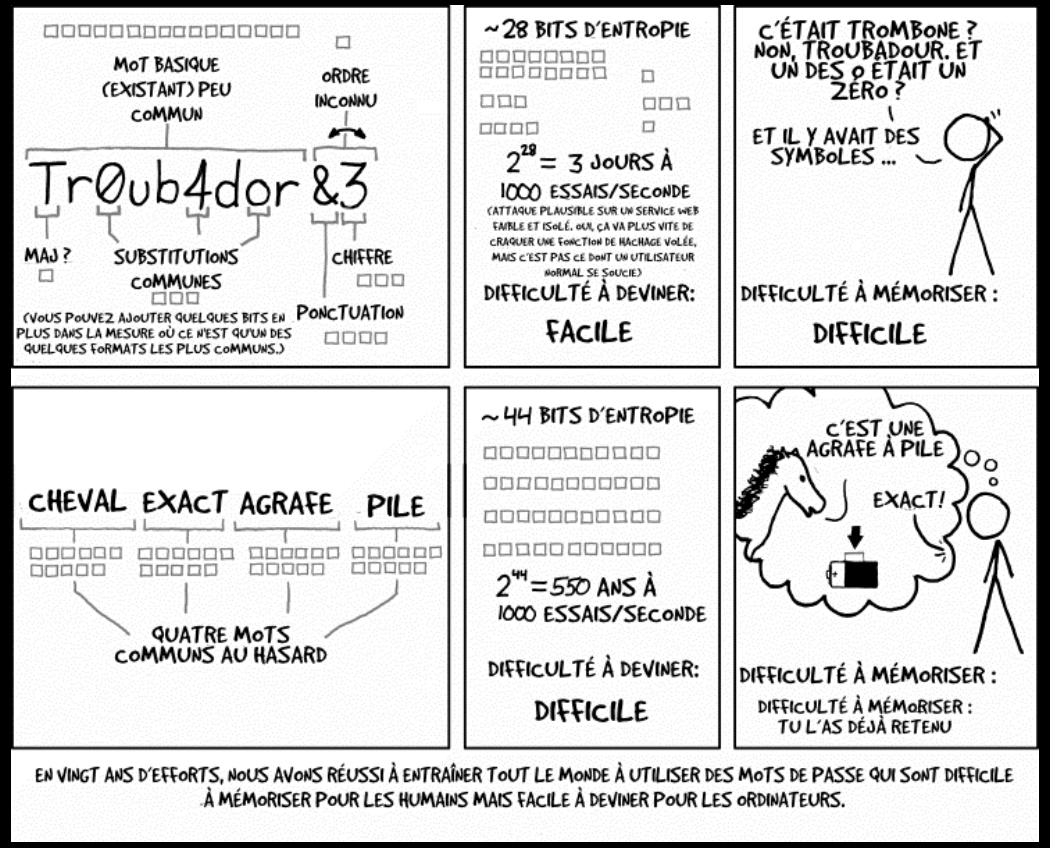
Résistance au brute force

Quantité d'information (de Shannon, en bits)

Limitations

Cognitives, psychologiques, …

XKCD



https://xkcd.lapin.org/index.php?number=936

OTP - One Time Password

Mot de passe à usage unique

(dans la pratique, on génère une liste)

e.g. ENIGMA

OTP - One Time Password

Basés sur le temps

```
hash(graine + timestamp / interval)
```

Liste chainée

```
hash(precedent)
```

Challenge / response

```
hash(graine + challenge)
```

Sessions

Suivre une activité répartie sur plusieurs requêtes

Sessions IP

« Identification Number »

(Défragmenter à l'arrivée)

Sessions TCP

« sequence » et « ack number »

(Acquittement des messages reçus)

Sessions TLS

Session ID

(paramètres déjà négociés)

Sessions Web

Identifiant dans un cookie

(PHPSESSID, JSESSID, …)

Vulnérabilités spécifiques

Stealing

(voler l'ID d'une victime)

Guessing

(deviner l'ID d'une victime)

Fixation

(forcer l'ID d'une victime)