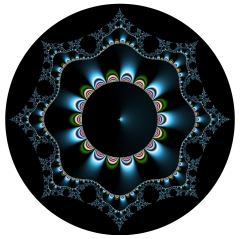
Sécurité des applications

Authentification et Contrôle d'accès

Thibaut et Corinne HENIN



www.arsouyes.org
@arsouyes

INSA
INSTITUT NATIONAL
DES SCIENCES
APPLIQUÉES
CENTRE VAL DE LOIRE

Sommaire

- Authentification
  - Contextuelle
  - Secret partagé
  - Clé publiques
  - Fournisseurs d’identité
  - Sessions
- Contrôles d’accès
  - Contournements
  - Modèle RBAC

I
Authentification

Qui est Qui ?

Facteurs d’authentification

- **Je sais**
  - Mot de passe
- **Je possède**
  - Téléphone, carte à puce, …
- **Je suis**
  - Biométrie
- **Je sais faire**
  - Signature manuscrite

Méthode d’authentification

- **Simple**
    - Un facteur d’authentification
- **Forte**
    - Au moins deux
    - *Contexte normatif*

Contextuelle

Adresse IP, HTTP Referer, HTTP Host, Cookies

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

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

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

Limitations sur les en-têtes

- Forgées par le client

Secret partagé

Mots de passes, OTP, multi facteurs

Mot de passe

- **Principe :**
    - Stocker l’association « identifiant / mot de passe »
    - L’utilisateur choisi et fourni son mot de passe

- **Sécurité spécifique :**
    - Stockage par le serveur
    - Choix par l’utilisateur

Mot de passe : Stockage

- **Level 0 :** en clair
    - identifiant / mot de passe

- **Attaques :**
    - Trivial ;-)

Mot de passe : Stockage

- **Level 1 :** hasher
    - Identifiant / hash(mot de passe)
    - Choix de la fonction (md5 *vs* sha512)

- **Attaques :**
    - Dictionnaires (*e.g.* google)
    - Tables arc-en ciel

Mot de passe : Stockage

- **Level 2 :** hashage + salage
    - Identifiant / sel + hash(mot de passe + sel)
    - Voir « bcrypt »

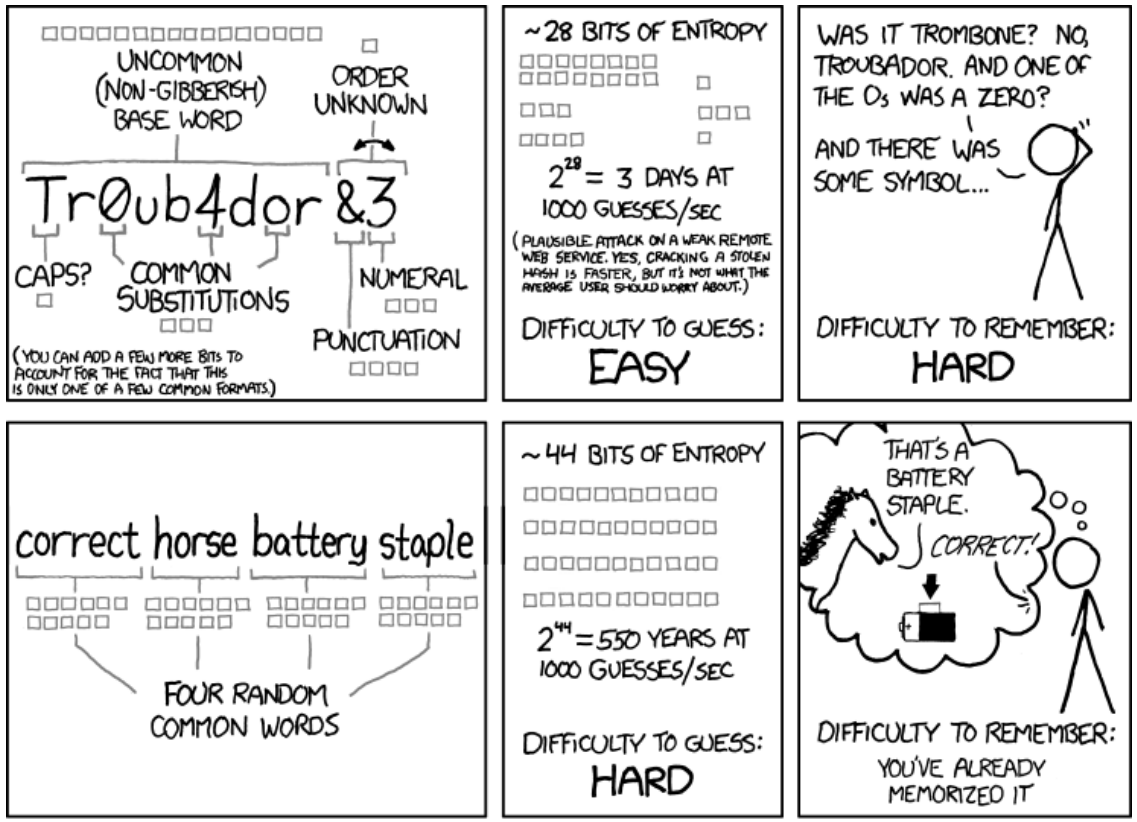
- **Attaques :**
    - Force brute
    - Dictionnaire

Mot de passe : choix

- **Entropie (de Shannon) :**
  - Quantité d'information (bits)
  - Taille de l'ensemble (2^bits)
  - Résistance au brute force

- **Problème :**
  - Limitations cognitives, psychologiques, …

XKCD


UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor&3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^28 = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS:
EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER:
HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^44 = 550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS:
HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER:
YOU'VE ALREADY MEMORIZED IT
THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

OTP - One Time Password

- **Principe :**
  - Mot de passe à usage unique
  - Générer une suite de mots de passes à partir

- **Exemple :**
  - Carnets de clés pour ENIGMA

OTP - One Time Password

- **Basés sur le temps :**
  - hash(graine + timestamp / interval)
- **Liste chainée :**
  - hash(precedent)
- **Challenge / response :**
  - hash(graine + challenge)
  - signature(clé privée, challenge)

Clé publique

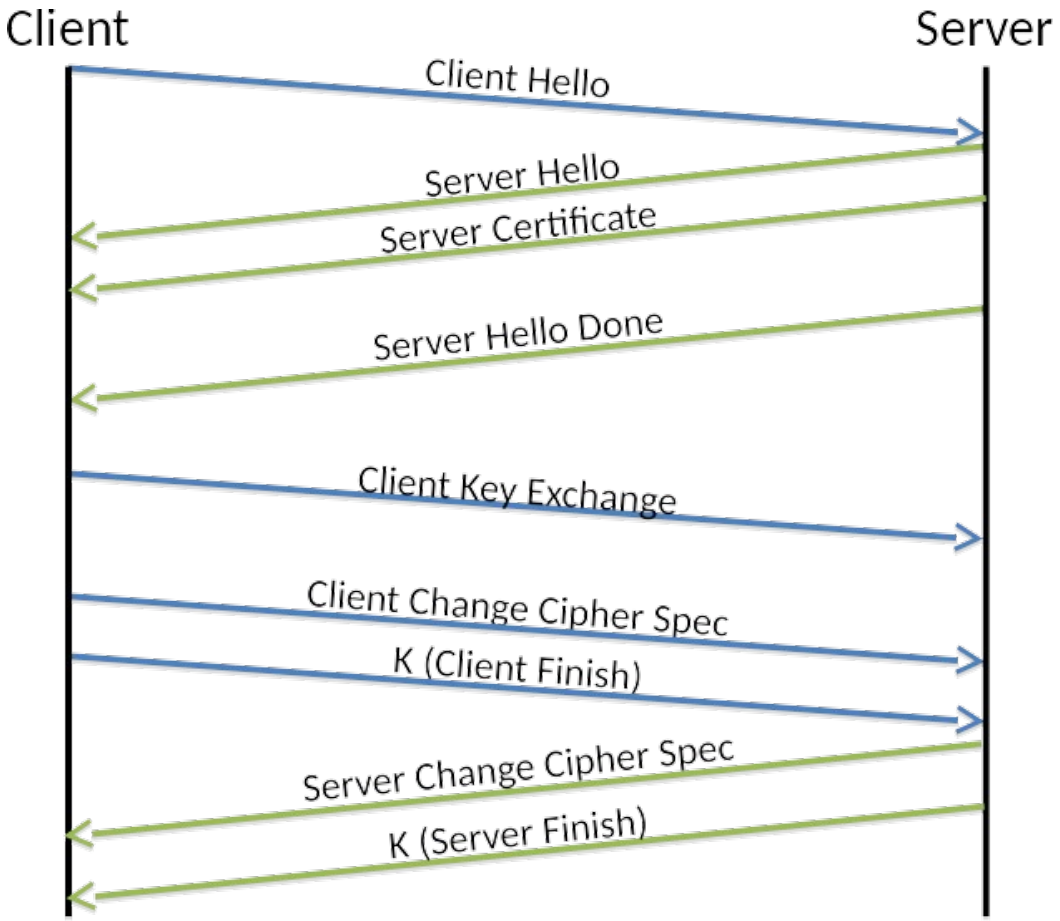
SSL/TLS, SAML, JWT

Rappel Cryptographiques

- **Cryptographie asymétrique**
    - Bi-clé (publique / privée)
    - Signature avec la clé privée

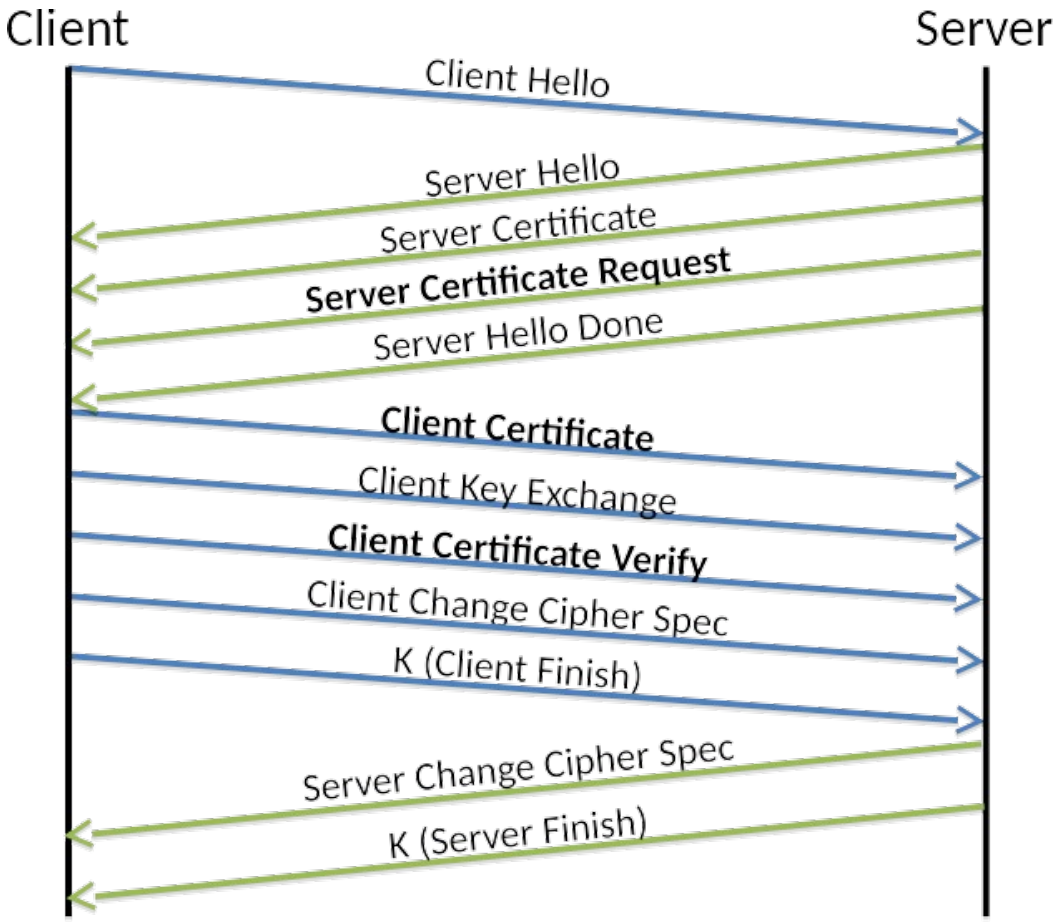
- **PKI – Public Key Infrastructure**
    - Certificat = clé publique + informations
    - Chaine de confiance par signatures

SSL/TLS : serveur



Source : http://www.allanbank.com/blog/security/tls/x.509/2014/10/13/tls-x509-and-mongodb/

SSL/TLS : mutuelle



Source : http://www.allanbank.com/blog/security/tls/x.509/2014/10/13/tls-x509-and-mongodb/

Signature des requêtes

- **Principe :**
  - La requête est signée avec la clé du client
  - Le serveur vérifie la signature

- **Exemples :**
  - **SAML -** Security assertion markup language
  - **JWT-** Json Web Token

Signature des requêtes : SAML

```
<samlp:Response ...>
  <saml:Issuer>http://idp.example.com/metadata.php</saml:Issuer>
  <samlp:Status>...</samlp:Status>

  <saml:Assertion ID="pfxcaa3deda-f4a7-863c-5d83-b714652c352c" ...>
    <saml:Issuer>http://idp.example.com/metadata.php</saml:Issuer>

    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
        <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
        <ds:Reference URI="#pfxcaa3deda-f4a7-863c-5d83-b714652c352c">...</ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>TYGJ1Z8+jGNpQuRcNAWTbk2.........En8IYtAUjsrSVsr4=</ds:SignatureValue>
      <ds:KeyInfo>
        <ds:X509Data>
          <ds:X509Certificate>MIICajCCAdOgAwIBAgIBAD........Gyc4LzgD0CROMASTWNg==</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </ds:Signature>

    <saml:Subject>...</saml:Subject>
    <saml:AuthnStatement ...>...</saml:AuthnStatement>
    <saml:AttributeStatement>
      <saml:Attribute Name="uid" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
        <saml:AttributeValue xsi:type="xs:string">test</saml:AttributeValue>
      </saml:Attribute>
      <saml:Attribute Name="mail" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
        <saml:AttributeValue xsi:type="xs:string">test@example.com</saml:AttributeValue>
      </saml:Attribute>
      <saml:Attribute Name="eduPersonAffiliation" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
        <saml:AttributeValue xsi:type="xs:string">users</saml:AttributeValue>
        <saml:AttributeValue xsi:type="xs:string">examplerole1</saml:AttributeValue>
      </saml:Attribute>
    </saml:AttributeStatement>
  </saml:Assertion>

</samlp:Response>
```

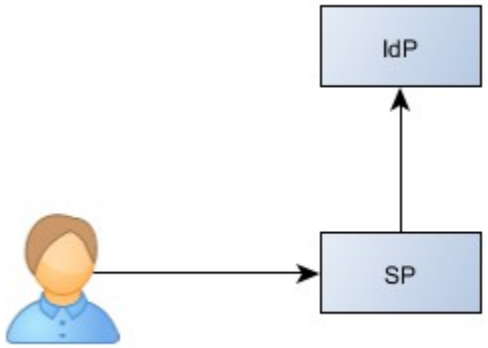
Fournisseurs d’Identité

oauth, openid, kerberos, ldap, …

Principe : Déléguation

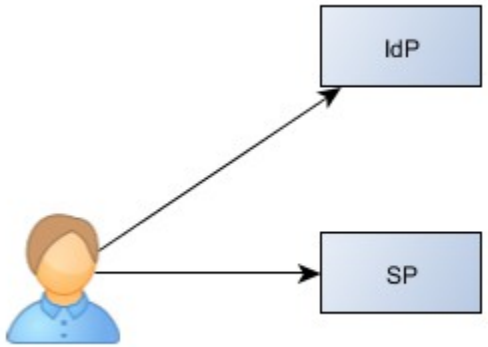
- **Fournisseur de Service**
    - « Service Provider / SP »
- **Fournisseur d’identité**
    - « Identity Provider » / « IdP »

Principe



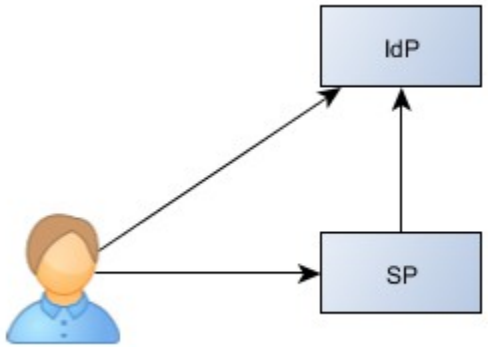
- .htpasswd
- Base de donnée
  - *e.g.* PHP + mySQL
- LDAP, Active Directory

Principe



- Kerberos
- SAML

Principe



- Oauth
- OpenID
- Tiers de paiement en ligne

Sessions

Suivre une activité répartie sur plusieurs requêtes

Sessions TCP/IP

- **IP :**
    - Défragmenter à l'arrivée
    - Champ « identification number »

- **TCP :**
    - Acquittement des messages reçus,
    - Champ « sequence » et « ack number »

Sessions Web

- **Identifiant de sessions**
    - **PHP :** PHPSESSID
    - **Tomcat :** JSESSID

- **Vulnérabilités :**
    - Stealing
    - Guessing
    - Fixation

Session : Stealing

- Vol de la session d'une victime
  - Man In the Middle
  - Scripts malveillants (XSS)
  - Partage du client (e.g. cyber-café)

- Protections :
  - Attribut « Secure » (interdit transfert en clair) + HTTPS
  - Attribut « HttpOnly » (interdit utilisation par les scripts)
  - Attribut « Domain » (interdit utilisation hors du site)
  - Attribut « SameSite » (interdit utilisation depuis un autre site)
  - Expiration automatique côté serveur

Session : Guessing

- Trouver un identifiant valide :
    - Brute Force
- Protection :
    - Générateur d’aléa sûr (i.e. cryptographique)
    - Entropie (i.e. 128 bits)

Session : Fixation

- Forcer l’identifiant d’une victime :
    - Créer une sessions personnelle
    - Forcer cet identifiant chez la victime
    - La session est alors partagée

- Protection :
    - Régénération de l’identifiant
        - Authentification (minimum)
        - Changement de rôles (avancé)

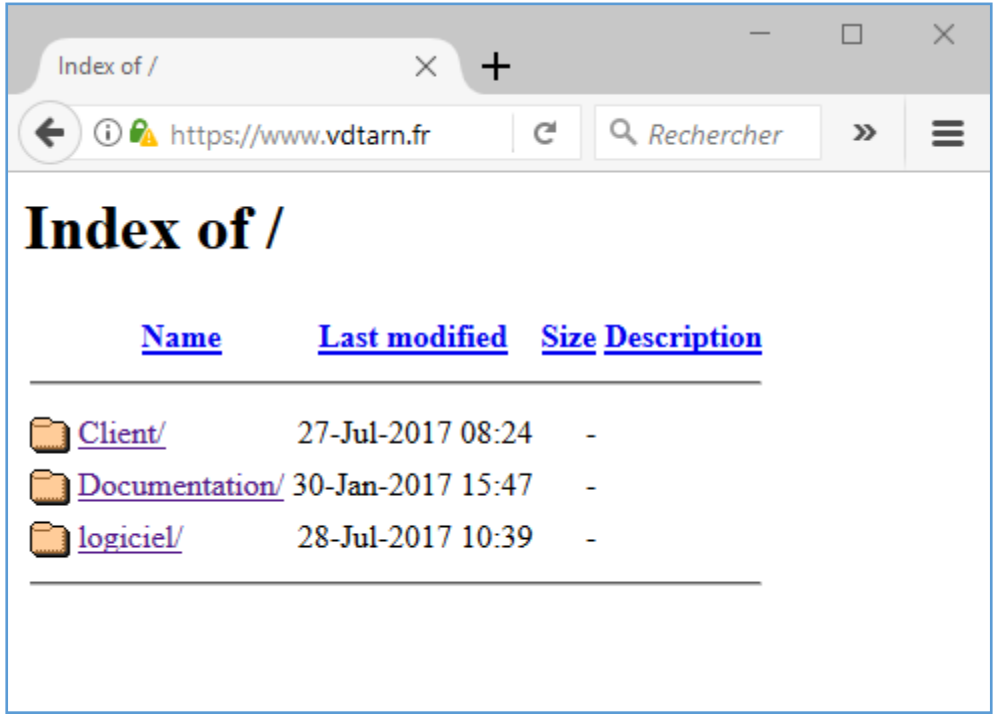
II
Contrôle d’accès

Contournements

Directory listing, Direct access (url, id, ...), Fopen

Directory listing

```
<VirtualHost *:80>

  ServerName  intranet.example.com

  Options +Indexes

</VirtualHost>
```



Direct access

```
$ wget http://intranet.example.com/upload/secret.pem

$ wget http://api.example.com/Keys/1f5d6s2d1
```

fopen

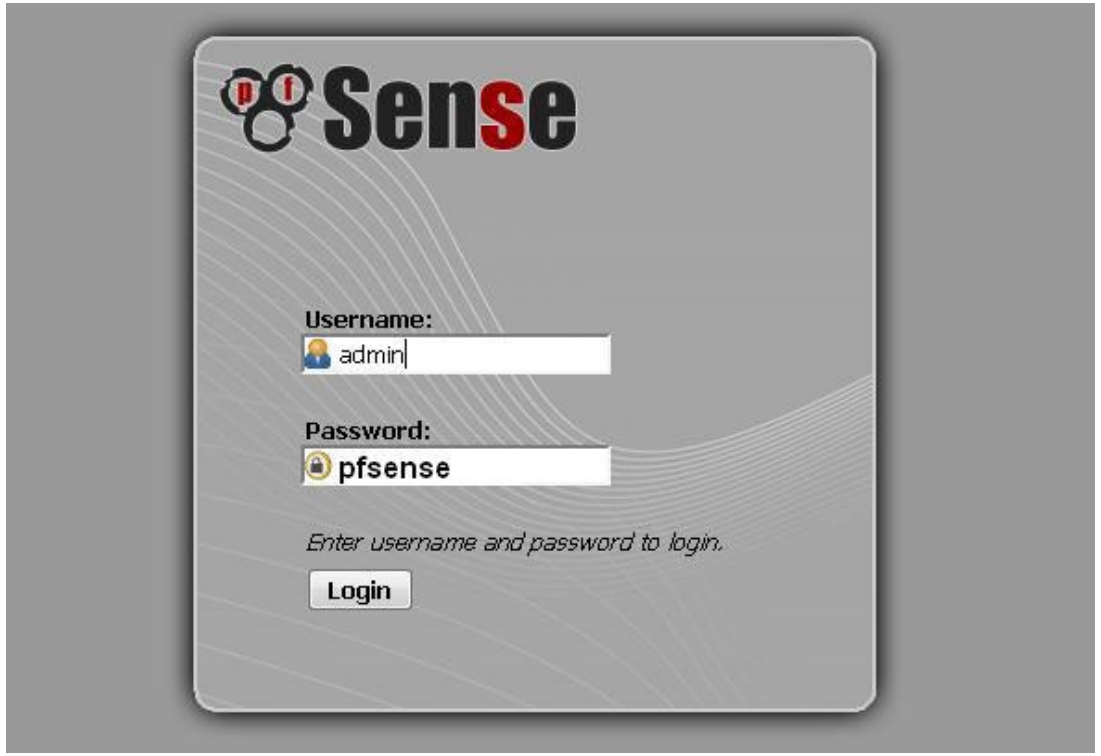
```
<?php

$file = "./some/dir/to/" . $_GET["file"] ;
$fp   = fopen($file) ;
$data = fread($fp, filesize($file)) ;
fclose($data) ;
echo $data ;

// readfile, file, file_get_contents, …
```

Featured Backdoors

Mode debug

```
<?php

$debug = isset($_GET["debug"]) ;
$user  = $_SESSION["user"] ;

if ($user->isAdmin() || $debug ) {

  do_admin_stuff($_GET, $_POST) ;

}
```

Compte admin



RBAC

Principe de base

- **Droits** – *permissions*
  - *Ce qui est contrôlé*
  - *E.g.* « *ajouter un utilisateur* »
- **Groupes** – *roles*
  - Ensemble de droits
  - *E.g. « administrateur »*
- **Utilisateurs** – *subjects*
  - Assignés à des groupes

Variantes

- Hiérarchique : héritage entre les rôles
- Contraint : un seul rôle a la fois

Principe du moindre privilège

- Droits **minimums** pour chaque groupe
- Groupes **minimums** pour chaque utilisateur

- Minimiser les risques

Le conseil des Arsouyes

Conseils des arsouyes

Authentification

- Chiffrez le réseau
- Salez les mots de passe
- Utilisez un framework

Contrôle d’accès

- Contrôlez tout
- Contrôlez tout le temps
- Faites des groupes