

Fonctionnement des PKI

- 2 -

**De la clef publique au
certificat**



Plan

Introduction

- × De la clef publique au certificat
- × Clef RSA, clef SSH, clef PGP
- × Problématique de la certification

Les certificats X.509

- × Standard X.509v3 et profil PKIX
- × Champs standards et extensions v3

En pratique

- × Panorama des certificats inclus dans IE 5.0
- × Sécurisation des échanges avec SSL/TLS



**De la clef publique
au
certificat**



Exemple de clef publique RSA (1/2)

- × Format « imprimable » PEM :

```
$ openssl x509 -in certificate.pem -pubkey -noout

-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAuRyXFwdfK4QgS7Yapk/D
SIVyHqBVN12VRSQy1A9y27f3IWAhAJyipGbZFDaPONpcwOZ376gdr4XLCRXD9nCA
BzNeEpIGBWjvaS8QyABoDuFo+kz01AT4WtAgVePqFydqNYV1eYiJk33B5oDW+zby
4Y5Ldtj014ktO9qjzhqZM94WvjlmMGafjbmV6QF8JWCl/kotKJr0FjDgXgCykbBY
ZgdPKCKgDp6lM3DC1OBSrB8+xlkIiJ5dFJKF1ClkfK0tQpz1B0UdCmhQ7+1Nw2jY
Ht+79jN8ULa7zAzlf+lbbFrJlhoB3YlR6FXf88HrlUGZVbLjX6NNOZIlZYbdYTx9
dwIDAQAB
-----END PUBLIC KEY-----
```

- × Format « humainement » lisible :

```
$ openssl x509 -in server.pem -pubkey | openssl rsa -text -noout -pubin
Modulus (2048 bit):
00:b9:1c:97:17:07:45:2b:84:20:4b:b6:1a:a6:4f:
[...]
c9:d6:1a:01:dd:89:51:e8:55:df:f3:c1:eb:95:41:
99:55:b2:e3:5f:a3:4d:39:92:25:65:86:dd:61:3c:
7d:77
Exponent: 65537 (0x10001)
```





Exemple de clef publique RSA (2/2)

Donnée brute, manipulable avec ambigüité sur :

- × Type de clef et d'usage
 - × Algorithme (RSA, DSA)
 - × Chiffrement, Signature, Échange de clef
 - × Tout autre usage spécifique à une application ?
- × Période de validité, statut de révocation
- × Identité (paramètre « administratif ») associée
 - × Adresse IP (192.0.2.1)
 - × Nom DNS pleinement qualifié (www.hsc.fr)
 - × RFC822 (franck.davy@hsc.fr)
- × Format à enrichir et à standardiser
 - À l'usage des applications



Première approche : SSH et dérivés (1/5)

- × Méthodes d'authentification
 - × Côté client
 - × Authentification par mot de passe (permanent ou à usage unique)
 - × Authentification par clef publique RSA ou DSA (SSHv2 uniquement)
 - × Côté serveur
 - × Authentification par clef publique (RSA/DSA)
 - × Clef publique transmise au client lors de la première ouverture de session
 - Sauvegardée côté client
- × Un format de clef minimaliste

```
$ cat ssh_host_key.pub  
  
1024 35 12312312312315016364622805541210905036255388057669  
8961735325312313245435645645671203023278678280118194562709  
6939937220982467588574628302525586302775483291130879549507  
8961735325312313245435645645671203023278678280118194562709  
3619651231243454354367454574568832342435454596754237631856  
093442973835538980270174901 root@darkstar
```



Exemple d'ouverture de session

- * Notion de « clef d'hôte »
- * Authentification du serveur est réalisée au niveau applicatif
- * Utilisation de clef RSA ou DSA
- * Objectif : *authentification* du serveur au niveau *applicatif* par sa clef publique
 - * Et non plus simplement *identification* par adresse IP ou FQDN
 - * Nombreuses attaques sur les protocoles IP, DNS et au niveau réseau...
 - Attaques de type « *Man-in-the-Middle* »
 - * DNS ID Spoofing, DNS Cache poisoning
 - * ProxyARP sur un réseau ethernet commuté (dsniff, ettercap)
- * Côté client

Connexion TCP (22/TCP)
Ouverture de session SSH



SSH_RSA_VERIFY

Non ? 5A:51:41:de:62:0c...

```
$ ssh darkstar
```

```
The authenticity of host 'darkstar (192.0.2.1)' can't be established.  
RSA key fingerprint is 5a:51:41:de:62:0c:d8:e9:c4:00:e1:19:f5:61:57:8c.  
Are you sure you want to continue connecting (yes/no)? Yes  
Warning: Permanently added 'server' (RSA) to the list of known hosts.
```

- * Côté serveur

```
$ ssh-keygen -l -f ssh_host_rsa_key.pub  
1024 5a:51:41:de:62:0c:d8:e9:c4:00:e1:19:f5:61:57:8c ssh_host_rsa_key.pub
```



Première approche : SSH et dérivés (3/5)

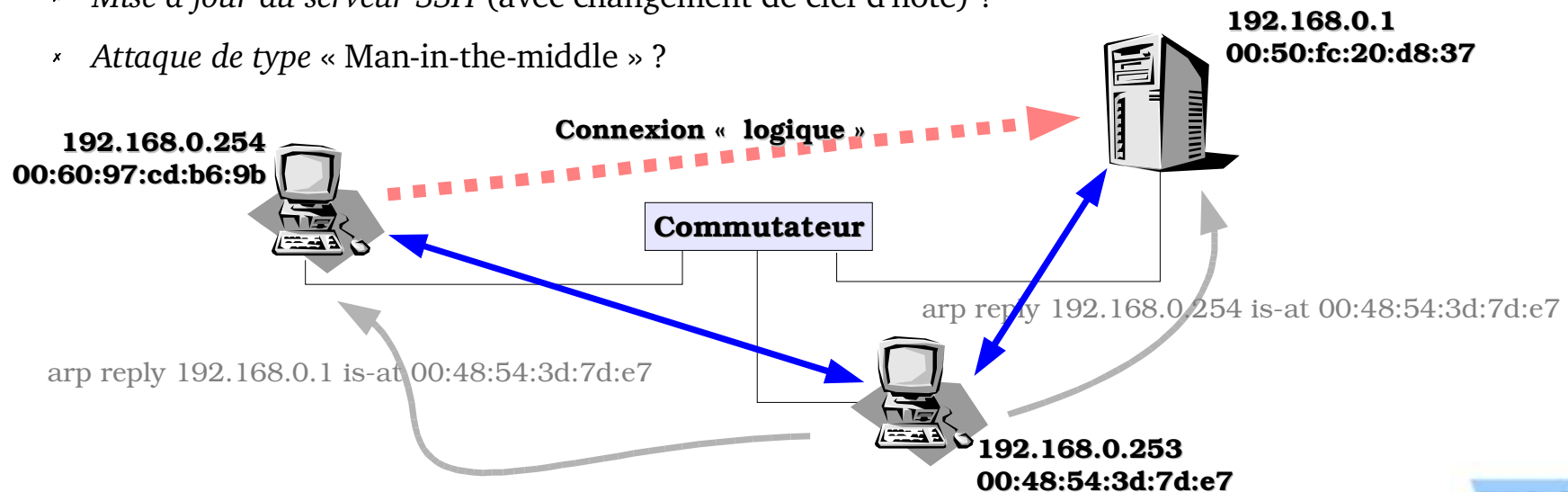
Avertissement lors d'un changement de clef publique du serveur

```

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@      WARNING: POSSIBLE DNS SPOOFING DETECTED!                  @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@      WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!       @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is:
5a:51:41:de:62:0c:d8:e9:c4:00:e1:19:f5:61:57:8c.
Please contact your system administrator.

```

- * Mise à jour du serveur SSH (avec changement de clef d'hôte) ?
- * Attaque de type « Man-in-the-middle » ?





Première approche : SSH et dérivés (4/5)

Problème de l'authenticité de la clef publique présentée

- × Contact de l'administrateur
 - Pour une vérification locale de l'empreinte de la clef présentée
- × Consultation d'un annuaire authentifié centralisant les clefs ou les empreintes
 - Exemple d'utilisation de DNSSEC
 - × Empreintes placées dans l'enregistrement de ressource TXT
 - Mais il s'agit déjà de PKI !*
- Nécessité d'un mécanisme de vérification hors-bande
 - × Administrateur système = rôle de tiers, garant de l'authenticité de la clef présentée
 - × Notion de « tiers de confiance »
 - × Tiers certifiant le lien entre clef publique et adresse IP ou un nom pleinement qualifié

Récapitulatif

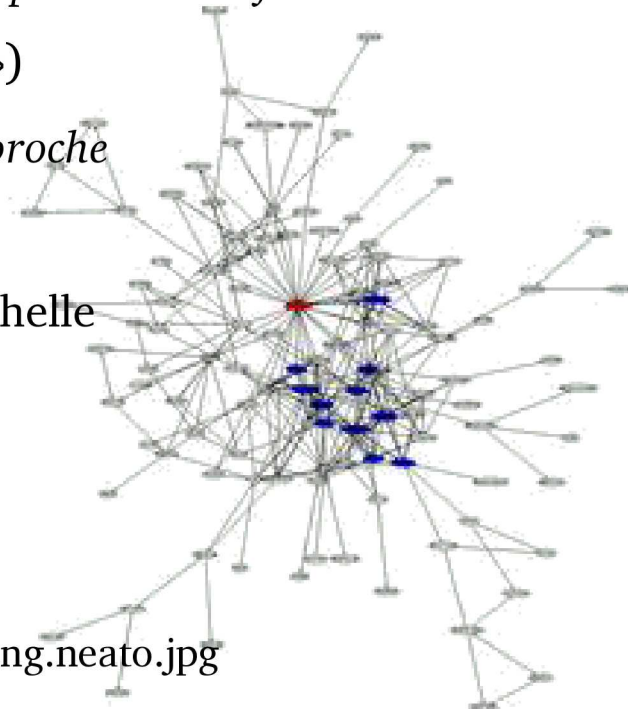
- × Intervention d'un tiers, dit « tiers de confiance »
- × Utilisation de mécanismes « hors-bande »
- × Première approche
 - × Introduction d'un *tiers de confiance*, certifiant les clefs publiques
 - Recours au *mécanisme de signature* de la cryptographie asymétrique
 - × Restriction des communications dites « hors-bande » avec ce tiers de confiance
 - Récupération de la clef publique de signature du tiers de confiance via un canal sûr
 - × Dans l'exemple précédent (changement de clef d'hôte SSH) :
 - × *Contact de l'administrateur système par téléphone*
 - × *Administrateur*
 - × *Préalablement connu de l'utilisateur*
 - *tiers de confiance*
 - × *Possédant un accès physique au système*



PGP et le « Web of Trust » (1/5)

Principe :

- × Cryptosystème hybride permettant l'échange chiffré/authentifié de données
 - Cryptographie à *clef secrète* pour le *chiffrement*
 - Cryptographie à *clef publique* pour la *signature*, le *transport* de la *clef session*
- × Principe de l'anneau de confiance (« *Web Of Trust* »)
 - × Authenticité d'une clef publique réalisée *de proche en proche*
- × Utilisation adaptée aux communautés...
 - × ... inadaptée aux échanges dématérialisés, à grande échelle
 - Notion de COI : « *Community of interest* »



Exemple : le « *Debian Keyring Web of Trust* »

- × <http://www.ChaosReigns.com/code/sig2dot/debian-keyring.neato.jpg>



PGP et le « Web of Trust » (2/5)

Distribution des clefs

- × Serveurs de clefs et serveurs HTTP référençant les clefs PGP

Envoi et recherche de clef sur un serveur de clef avec gpg :

```
$ gpg -keyserver hkp://wwwkeys.pgp.net:11371 --send-keys <keyid>
$ gpg -keyserver hkp://wwwkeys.pgp.net:11371 --recv-keys <keyid>
```

Recherche d'une clef suivant une chaîne de caractères :

```
$ w3m 'http://pgpkeys.pgp.net/
Public Key Server -- Index ``hsc.fr ''
Type bits /keyID Date User ID
pub 1024D/07ACF6FA 2001/11/22 Thomas Seyrat <Thomas.Seyrat@hsc.fr>
pub 1024D/34AFBB17 2001/04/18 Jerome Poggi (Office Key) <jerome.poggi@hsc.fr>
pub 1024D/B068C137 1999/10/15 Alain Thivillon <Alain.Thivillon@hsc.fr>
pub 1024R/D1D602E3 1999/02/26 Denis Ducamp <Denis.Ducamp@hsc.fr>
pub 1024R/0F2C58BD 1997/07/02 Stephane Aubert <aubert@hsc.fr>
pub 1024R/57155CC9 1996/11/09 Alain Thivillon <Alain.Thivillon@hsc.fr>
pub 1024R/D4ED2595 1995/12/08 Herve Schauer <Herve.Schauer@hsc.fr.net>
```

Exemple d'une recherche menée sur le nom *Zimmerman* sur le dépôt `keyserver.pgp.com`

```
Public Key Server - Error
Search failed
The number of certificates returned by this search exceeds
the maximum set for this server.
```





PGP et le « Web of Trust » (3/5)

Solutions retenues en pratique

- × Téléchargement à une URL, en HTTP par exemple
 - × Empreinte diffusée par courrier électronique (en-tête SMTP : X-GPG-Fingerprint)
- × « GnuPG Party »
 - × *Procédures décrites dans le « GnuPG Keysigning Party » HOW-TO*





PGP et le « Web of Trust » (4/5)

- * Format d'une clé PGP
 - * Extrait avec PGPDump

```
$ ./pgpdump ~/.gnupg/Franck.Davy.asc
[...]
```

Old: Public Key Packet(tag 6)(418 bytes)
Ver 4 - new
Public key creation time - Fri Aug 31 16:52:54 CEST 2001
Pub alg - DSA Digital Signature Standard(pub 17)
DSA p(1024 bits) - ...
DSA q(160 bits) - ...
DSA g(1024 bits) - ...
DSA y(1020 bits) - ...

```
Old: User ID Packet(tag 13)(32 bytes)
User ID - Franck DAVY <Franck.Davy@hsc.fr>
```





PGP et le « Web of Trust » (5/5)

* « GnuPG Party »

* Procédures décrites dans le « GnuPG Keysigning Party » HOW-TO

What do I need for this party?

Required Items

1. Physical attendance
2. Positive picture ID
3. Your Key ID, Key type, HEX fingerprint, and Key size
4. A pen/pencil or whatever you'd like to write with....
5. NO computer

Why shouldn't I bring a computer?

- * Someone might have modified the computers programs, operating system, or hardware to steal or modify keys.

* Format GnuPG Party des clefs

Key ID	Key Owner	Key Fingerprint	Key Size	Key Type
06E9EB78	Franck DAVY <Franck.Davy@hsc.fr>	AD11 1F76 25E2 5614 B07B EE22 ECA6 AE99 06E9 EB78	1024	DSA



Les certificats X.509



Définition

Un certificat est un lien de confiance entre :

- × Une identité
- × Une clef publique
- × Une période de validité
- × Un usage

Typologie des infrastructures à clef publique (PKI):

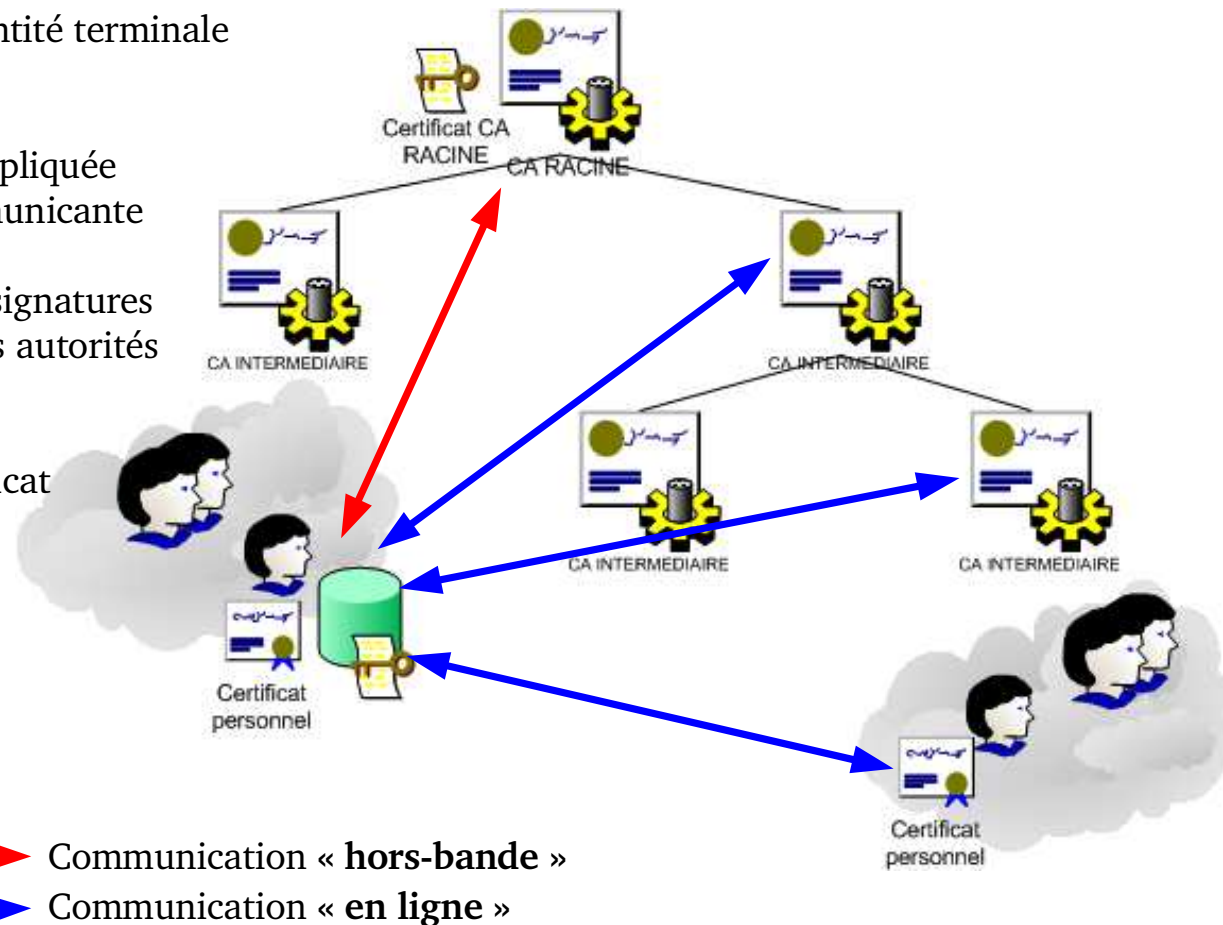
- × Une structure en théorie *hierarchique*
 1. Au sommet : une autorité de certification dite « racine »
 - × Une autorité en laquelle la communauté d'utilisateurs a confiance
 - × Certificat auto-signé
 - × Extension par certification croisée...
 2. Autorités de certification intermédiaires
 - × Certificats délivrés par l'autorité racine
 - × Habilitées, par exemple, à délivrer des certificats aux entités terminales
 3. Entités terminales (Serveur SSL/TLS, client S/MIME etc.)



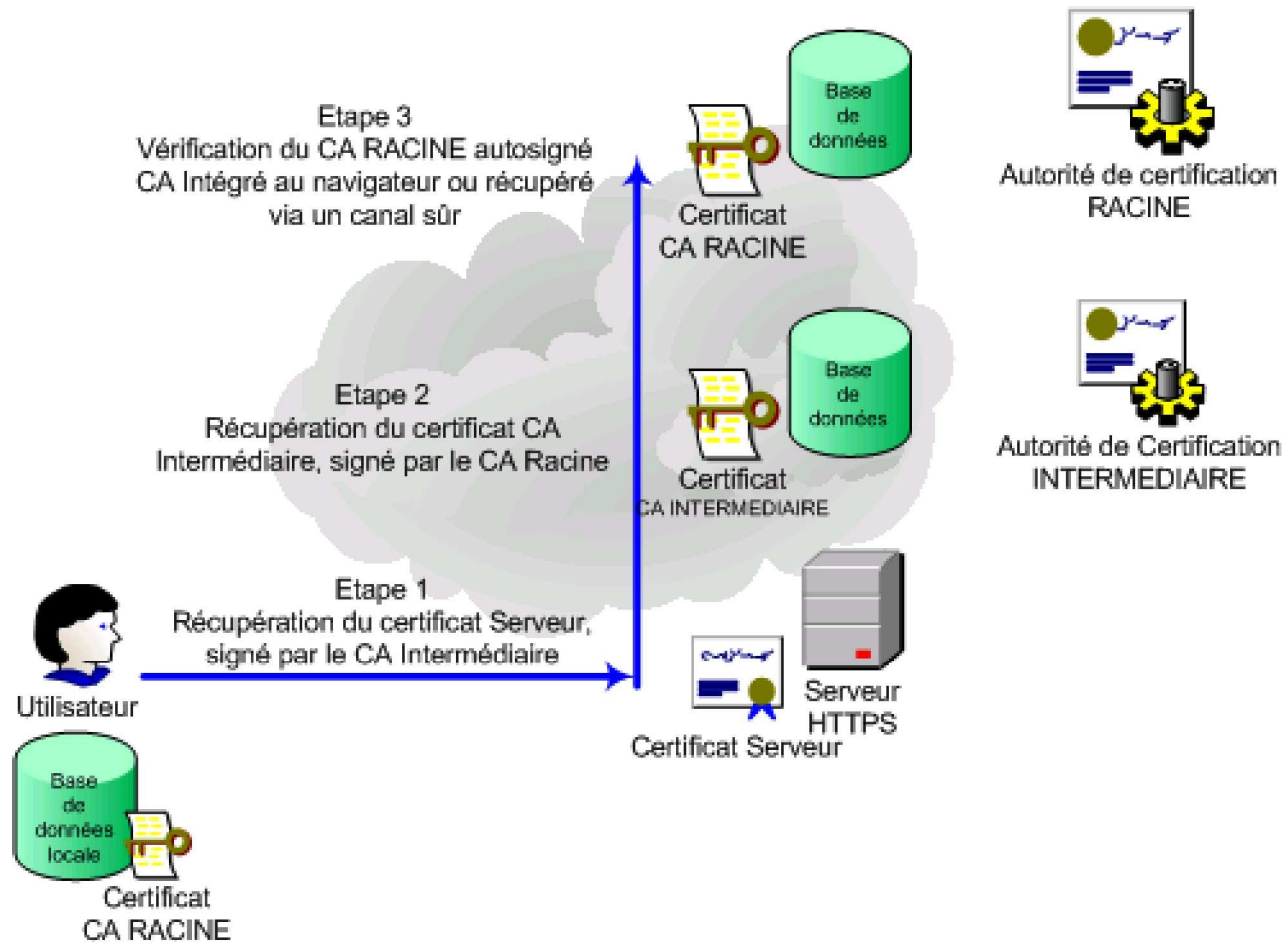
Structure hiérarchique

La validation du certificat d'une entité terminale est réalisé de **proche en proche** :

1. Validation de la signature appliquée au certificat de l'entité communicante
2. Validations successives des signatures appliquées aux certificats des autorités intermédiaires
3. Confiance accordée au certificat de l'autorité racine....



Exemple : validation d'un certificat serveur SSL/TLS





Itinéraire de certification (3/9)

s:/C=US/O=VeriSign, Inc./OU=Class 3 Public Primary Certification Authority

i:/C=US/O=VeriSign, Inc./OU=Class 3 Public Primary Certification Authority

s:/O=VeriSign Trust Network/OU=VeriSign, Inc./OU=VeriSign International Server CA - Class 3
/OU=www.verisign.com/CPS Incorpor.by Ref. LIABILITY LTD.(c)97 VeriSign

i:/C=US/O=VeriSign, Inc./OU=Class 3 Public Primary Certification Authority

s:/C=FR/ST=Paris/L=Paris/O=CERTPLUS SA/OU=Operations - 2003
/OU=Terms of use at www.certplus.com/rpa (c) 01/OU=Authenticated by Certplus SA
/OU=Member, VeriSign Trust Network/CN=www.certplus.com

i:/O=VeriSign Trust Network/OU=VeriSign, Inc./OU=VeriSign International Server CA - Class 3
/OU=www.verisign.com/CPS Incorpor.by Ref. LIABILITY LTD.(c)97 VeriSign





Itinéraire de certification (4/9)

En pratique :

- × Certificat = structure certifiée
 - × Ensemble clef publique + paramètres administratifs scellé par la clef privée de l'autorité signataire
- × Pour les différentes entités
 - × Possession du certificat de l'autorité racine, uniquement
 - × Parcours de l'itinéraire de certification (suivant les applications...) pour valider la chaîne des certificats présentés
- × Organisation hiérarchique ?
 - × Multiples autorités de certification
 - × 106 autorités de certification, ou autorités dites de « confiance » dans Windows 2000
 - × Passage d'une organisation du type « *hierarchy of trust* » vers une (dés)organisation « *spaghetti of doubt* » (Peter Gutmann)
 - × À la charge des applications de permettre au client de vérifier le certificat présenté
 - × Présentation de la chaîne de certification dans les protocoles SSL/TLS, S/MIME



Validation d'un certificat serveur SSL/TLS

- × Cas d'un serveur « correctement » configuré
 - × Présentation au client de l'intégralité de la chaîne de certification
 - × Pour le logiciel client :
 - × Pas de manipulations supplémentaires pour le téléchargement des certificats intermédiaires
 - × Possession préalable du certificat de l'autorité signataire requise, uniquement



- × Cas d'un serveur mal configuré
 - × Les différents certificats de la chaîne ne sont pas présentés au client
Le logiciel client doit-il la compléter ?
Explication de la présence de certificats *intermédiaires* dans IE ? (leur importation n'est pas gratuite !)
- × En général :
 - × Certificats terminaux, émis par une autorité racine (`pathlen = 0`)



Itinéraire de certification (6/9)

Validation SSL/TLS

```
$ openssl s_client -connect www.webserver.com:443 -CAfile ca/caroot.pem -showcerts
CONNECTED(00000003)
---
Certificate chain
0  s:CN=www.webserver.com
   i:CN=CA SSL
-----BEGIN CERTIFICATE-----
[...]
-----END CERTIFICATE-----
1  s:CN=CA SSL
   i:CN=CA ROOT
-----BEGIN CERTIFICATE-----
[...]
-----END CERTIFICATE-----
2  s:CN=CA ROOT
   i:CN=CA ROOT
-----BEGIN CERTIFICATE-----
[...]
-----END CERTIFICATE-----
Server certificate
    subject=/CN=www.webserver.com
    issuer=/CN=CA SSL
```





Itinéraire de certification (7/9)

Validation SSL/TLS

```
SSL handshake has read 4104 bytes and written 268 bytes
--
New, TLSv1/SSLv3, Cipher is EDH-RSA-DES-CBC3-SHA
Server public key is 1024 bit
SSL-Session:
  Protocol      : TLSv1
  Cipher       : EDH-RSA-DES-CBC3-SHA
  Session-ID:  269BC84135E313F12A9E624065FAC6FA9452D2F728AE518FEB17E009B8E3FF8B
  Session-ID-ctx:
  Master-Key:  95F39BE9741D3B55B418A3B6BC8B0F9B3649CF \
                F629B2259DEA36C4EAF16D4AB011FE6F62D914A679F7E915344AE8938E
  Key-Arg      : None
  Start Time:  1008002520
  Timeout      : 300 (sec)
  Verify return code: 0 (ok)
---
GET / HTTP/1.0 200 ok
Content-type: text/html[...]
```

- × Seul le certificat de l'autorité racine était initialement connu du navigateur
 - ➔ La validation a été correctement effectuée, par vérification *de proche en proche* des certificats constituant la chaîne de certification

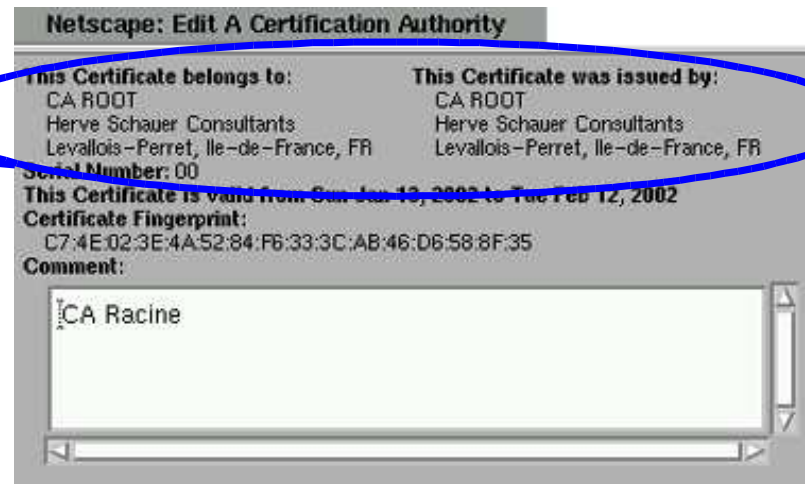




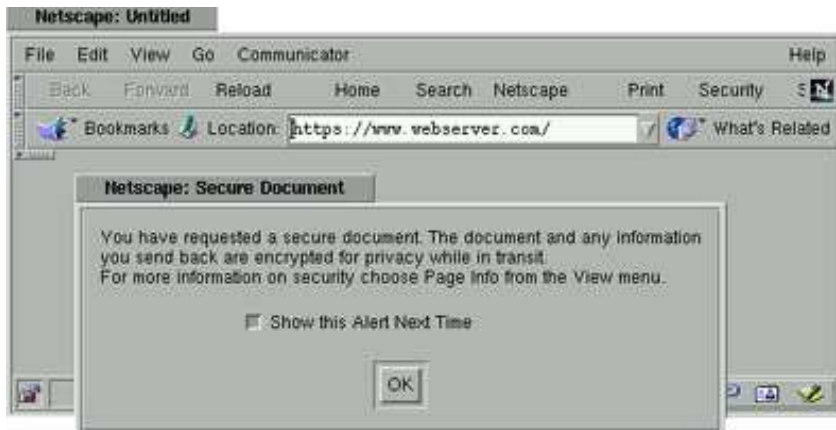
Itinéraire de certification (8/9)

Validation avec Netscape

- × Présent dans Netscape : le certificat de l'autorité racine uniquement
 - × CA ROOT autosigné
 - × DN du porteur = DN de l'émetteur



Validation avec Netscape correctement effectuée



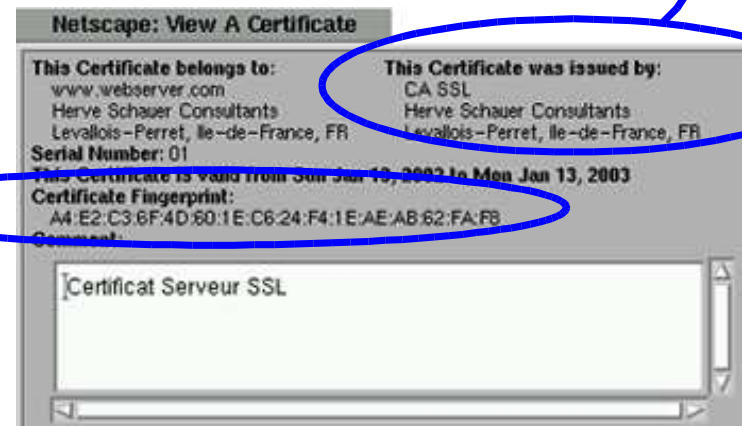
L'émetteur du certificat serveur est CA SSL

→ Seul le certificat de CA ROOT (CN=CA ROOT) figurait dans la base de données locale des certificats dits « *de confiance* »

Présentation de l'empreinte du certificat calculée par le navigateur

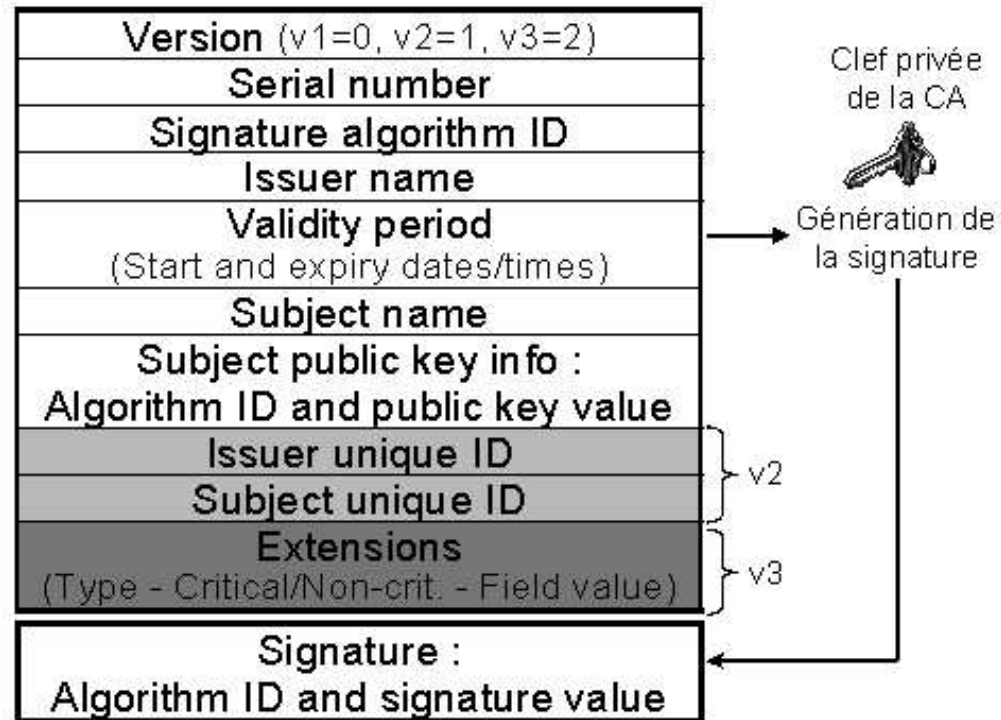
→ Empreinte MD5 du certificat au format DER

Analogue au mécanisme employé par OpenSSH, lorsque la vérification à partir de la clef du CA n'est pas réalisable



Historique

- × Norme ITU-T X.509, fait partie des recommandations X.500
- × Version 1 (1988) :
Définition des champs de base
- × Version 2 (1993) :
Ajout de 2 champs optionnels
- × Version 3 (1996) :
Ajout d'extensions
 - Avec notion de *criticité*
 - Importance pour la sécurité !
 - × basicConstraints
 - × keyUsage
 - × Etc.





Format X.509 (2/5)

Structure ASN.1

- * Abstract Syntax Notation 1
- * Syntaxe abstraite normalisée (X.208)
- * Encodage : syntaxe de transfert distinctive
 - Règle d'encodage DER

```
certificate ::= SEQUENCE {
    tbsCertificate      TBSCertificate,
    signatureAlgorithm  AlgorithmIdentifier,
    signatureValue      BIT STRING
}
```

```
TBSCertificate ::= SEQUENCE {
    version             [0] EXPLICIT Version DEFAULT v1,
    serialNumber        CertificateSerialNumber,
    signature            AlgorithmIdentifier,
    issuer              Name,
    validity            Validity,
    subject             Name,
    subjectPublicKeyInfo SubjectPublicKeyInfo,
    issuerUniqueID      [1] IMPLICIT UniqueIdentifier OPTIONAL,
                        -- If present, version shall be v2 or v3
    subjectUniqueID     [2] IMPLICIT UniqueIdentifier OPTIONAL,
                        -- If present, version shall be v2 or v3
    extensions          [3] EXPLICIT Extensions OPTIONAL
                        -- If present, version shall be v3
}
```





Format X.509 (3/5)

Certificate:

Data:



tBSCertificate

Version: 1 (0x0)

Serial Number:

32:50:33:cf:50:d1:56:f3:5c:81:ad:65:5c:4f:c8:25

Signature Algorithm: md2WithRSAEncryption

Issuer: C=US, O=VeriSign, Inc., OU=Class 1 Public Primary Certification Authority

Validity

Not Before: Jan 29 00:00:00 1996 GMT

Not After : Jan 7 23:59:59 2020 GMT

Subject: C=US, O=VeriSign, Inc., OU=Class 1 Public Primary Certification Authority

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Modulus (1024 bit):

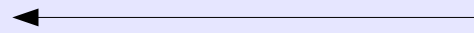
00:e5:19:bf:6d:a3:56:61:2d:99:48:71:f6:67:de:

[...]

2a:2f:31:aa:ee:a3:67:da:db

Exponent: 65537 (0x10001)

Signature Algorithm: md2WithRSAEncryption



signatureAlgorithm

4b:44:66:60:68:64:e4:98:1b:f3:b0:72:e6:95:89:7c:dd:7b:

[...]



signatureValue

f8:45





Format X.509 (4/5)

- × Format texte du champ identifiant l'émetteur

```
Issuer: C=US, O=VeriSign, Inc., OU=Class 1 Public Primary Certification Authority
```

- × Représentation ASN.1

```
: . . . SEQUENCE {
: . . . . SET {
: . . . . . SEQUENCE {
: . . . . . . OBJECT IDENTIFIER countryName (2 5 4 6)
: . . . . . . PrintableString 'US'
: . . . . . . }
: . . . . . . }
: . . . . . SET {
: . . . . . . SEQUENCE {
: . . . . . . . OBJECT IDENTIFIER organizationName (2 5 4 10)
: . . . . . . . PrintableString 'VeriSign, Inc.'
: . . . . . . . }
: . . . . . . . }
: . . . . . . SET {
: . . . . . . . SEQUENCE {
: . . . . . . . . OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
: . . . . . . . . PrintableString
: . . . . . . . . 'Class 1 Public Primary Certification Authority'
: . . . . . . . . }
: . . . . . . . . }
: . . . . . . . }
: . . . . . . }
: . . . . . }
```





Format X.509 (5/5)

- * Structure ASN.1 pour l'attribut 'issuer'

```
Name ::= CHOICE {
    RDNSequence }
RDNSequence ::= SEQUENCE OF RelativeDistinguishedName
RelativeDistinguishedName ::= SET OF AttributeTypeAndValue
AttributeTypeAndValue ::= SEQUENCE {
    type      AttributeType,
    value     AttributeValue }
AttributeType ::= OBJECT IDENTIFIER
AttributeValue ::= ANY DEFINED BY AttributeType
```

- * OID pour le RDN Country (C)

```
2.5.4.6 - id-at-countryName
countryName ATTRIBUTE ::= {
    SUBTYPE OF name
    WITH SYNTAX PrintableString (SIZE (2))
    SINGLE VALUE TRUE
    ID id-at-countryName
}
```

Avec :

```
2.5.4 - X.500 attribute types
2.5 - X.500 Directory Services
2 - ISO/ITU-T jointly assigned OIDs
Top of OID tree
```





Profil PKIX (1/2)

Description

- × Groupe de travail IETF créé en 1995
- × Objectif : développer pour l'internet une PKI fondée sur les certificats X.509

Composantes

- × *Instanciación des certificats X.509v3 et des listes de révocations X.509 pour une infrastructure adaptée à l'Internet (RFC3280)*
- × *Protocoles d'exploitation (RFC2559, RFC2585)*
 - × Distributions des certificats et listes de révocation
- × *Protocoles de gestion (RFC2510)*
 - × Dialogues entre les différentes entités de la PKI
- × *Règles d'usage et considérations pratiques (RFC2527)*
 - × Exigences de sécurité
 - × En matière d'identification des sujets, de révocation des certificats



Profil PKIX (2/2)

Existence de nombreux profils...

- × PKIX, FPKI (US Federal PKI Profile), MISSI (US DoD profile), ISO 15782, SEIS (Secured Electronic Information For Society), Australian Profile, German Profile, Microsoft Profile etc.

« You can' be a real country unless you have a beer and a airline.
It helps if you have some kind of a football team, or some nuclear
weapons, but at the very least, you need a beer.»

-- Franck Zappa

« And an X.509 profile. »

-- Peter Gutmann

- × Difficultés dans le choix des produits (et prestataires)
 - × Conformité au profil PKIX requise ?
 - × Ex : Microsoft - Ignorance du bit keyUsage
 - × ... et de l'extension basicConstraints ! (Cf. Faille CryptoAPI)
 - × Pour aller plus loin : « *X.509 Style Guide* »
 - × <http://www.cs.auckland.ac.nz/~pgut001/>





Champs standards et X.500 (1/3)

Version

- × Indique l'utilisation possible d'extensions
- Problème de compatibilité avec les extensions propriétaires

```
X509v3 extensions:  
2.5.29.1:  
0i.....[...*.9.b.S2.R0P1.0...U...US1^M0...U.  
..MSFT1200..U...)Microsoft Authenticode(tm) Root Authority...
```

- × *En pratique* : version 3

Numéro de Série

- × Identifiant unique d'un certificat (pour une autorité signataire donnée)
- × *En pratique* :
 - × Incrémenté aléatoirement (estimation du nombre de certificats émis ?)

Champs "Subject" et "Issuer"

- × Comportent respectivement les Distinguished Names (DN) de l'émetteur et du signataire du certificat
 - Pour le certificat d'une autorité de certification racine, ces deux champs sont égaux
- × En théorie, constitue un chemin unique vers l'entité possédant le DN référencé dans le DIT





Champs standards et X.500 (2/3)

- × *En pratique* : inexistence d'annuaire X.500...
 - × Peu de sens en pratique (sauf contexte LDAP !)
 - × Seul le commonName est généralement utilisé
 - × Cas des serveurs HTTPS notamment
 - × Tous les RDN peuvent importer dans le cas d'un certificat client
 - Gestion des autorisations par exemple
 - × Les RDN sont généralement sans grand rapport avec leur fonction originale

```
Issuer= /O=VeriSign Trust Network  
/OU=VeriSign, Inc.  
/OU=VeriSign International Server CA - Class 3  
/OU=www.verisign.com/CPS Incorpor.by Ref. LIABILITY LTD.(c)97 VeriSign
```

- × Recommendation PKIX : utiliser l'extension v3 subjectAltName
 - Gestion des alias, pour les hôtes virtuels, ou encore les adresses email multiples dans le cas du courrier électronique chiffré/signé au format S/MIME





Champs standards et X.500 (3/3)

Validité

- * Dates déterminant la période de validité d'un certificat
 - * Au delà de cette période, on parle d'*expiration*
 - Suspension de validité durant cette période : *révocation*
- * Remarques :
 - * Certificats révoqués arrivés à expiration susceptibles de rester dans les CRL
 - * Propice à l'accroissement en taille des CRL...
 - * Extrait de l'archive de CRL <http://crl.verisign.com>

```
$ ls -al RSAsecureServer*  
-rw----- 1 davy davy 822074 sep 20 09:01 RSAsecureServer.crl  
-rw----- 1 davy davy 841986 sep 20 09:01 RSAsecureServer.crl.0710  
-rw----- 1 davy davy 766740 sep 20 09:01 RSAsecureServer-p.crl.old  
-rw----- 1 davy davy 768420 sep 20 09:01 RSAsecureServer-p.crl.save
```

- * Cependant certains navigateurs ne vérifient pas correctement ces dates (Netscape)
 - * Exemple : Applet Java signée... mais certificat expiré
 - * [Http://www.brookscole.com/compsci/aeonline/course/7/2/index.html](http://www.brookscole.com/compsci/aeonline/course/7/2/index.html)





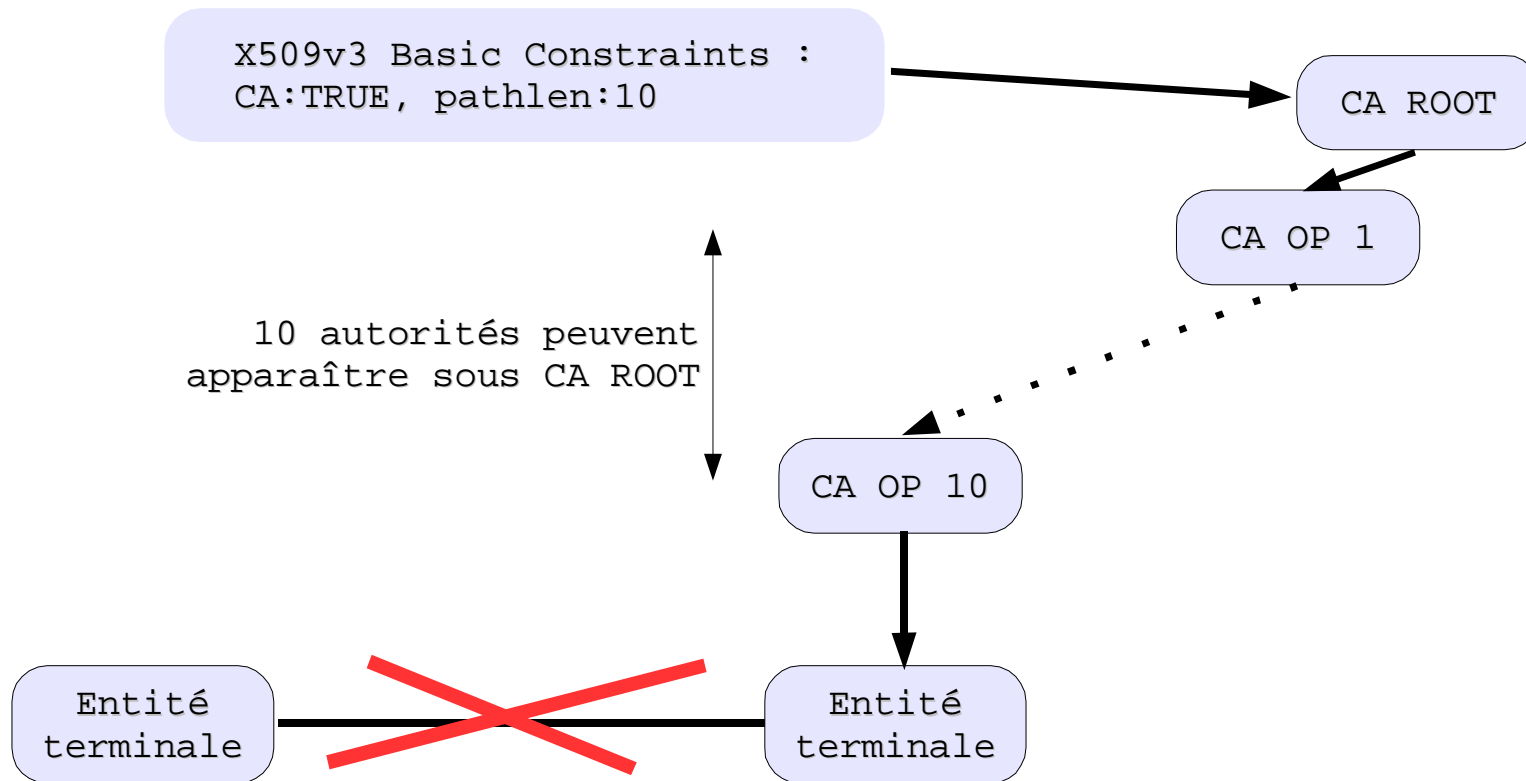
Extensions X.509v3 (1/8)

- × Notion de **criticité** d'une extension
- × **Contraintes** d'utilisation, sur l'**usage**
 - Clef publique de chiffrement (RSA), de signature (RSA/DSA), d'échange de clef (DH)?
 - Exigences de sécurité
 - × *Service de non répudiation associé ?*
- × Contraintes sur l'itinéraire de certification
 - Longueur maximale de la chaîne de certification
- × Informations diverses
 - × Points de distribution des listes de révocation, de la politique de certification etc.
 - × Extensions propriétaires
- × *En pratique :*
 - × Considération laissée à la discrétion de l'application
 - × Peu d'extensions reconnues ou prises en compte
 - × Avec de potentielles failles de sécurité en conséquence
 - CryptoAPI MS02-050



basicConstraints

- × Pathlen : nombre de *certificats* pouvant apparaître, dans la chaîne de certification, *sous* le certificat portant cette extension
- × CA : TRUE pour une entité habilitée à délivrer des certificats, FALSE sinon





Extensions X.509v3 (3/8)

keyUsage

- × `digitalSignature` (RSA/DSA)
 - × mécanisme de signature
- × `nonRepudiation` (RSA/DSA)
 - × service de non répudiation
- × `keyEncipherment/dataEncipherment` (RSA)
 - × chiffrement de clef (de session, premastersecret) ou de données
- × `keyCertSign/cRLSign` (RSA/DSA)
 - × signature de certificats/de listes de révocation
- × `keyAgreement/encipherOnly/decipherOnly` (DH)
 - × utilisation de Diffie-Hellman uniquement

X509v3 Key Usage:
Certificate Sign, CRL Sign





Extensions X.509v3 (4/8)

- × Variantes autour de l'extension keyUsage
 - × Extended Key Usage (extendedKeyUsage)
 - × serverAuth, clientAuth, codeSigning, emailProtection, timeStamping, msCodeInd, msCodeCom, msCTLSign, msSGC, msEFS, nsSGC
 - × Netscape Cert Type (nsCertType)
 - × objsign, email, server, objCA, emailCA, sslCA

X509v3 Extended Key Usage:

TLS Web Client Authentication, E-mail Protection

Netscape Cert Type:

SSL CA, S/MIME CA, Object Signing CA





Extensions X.509v3 (5/8)

cRLDistributionPoints

- × URI ou DNS: point de téléchargement de la liste de révocation correspondant au certificat portant cette extension
- × DirName (DN) permettant de télécharger la liste de révocation via LDAP

X509v3 CRL Distribution Points:

URI:http://crl-acracine.certinomis.com/acracine.crl

DNS:ldap.certinomis.com

DirName:/C=FR/O=CertiNomis/OU=AC Racine - Root CA/CN=CertiNomis

- × Recherche de la CRL via ldapsearch(1)
 - × Suite d'outils OpenLDAP
<http://www.openldap.org>

```
# CertiNomis Classe 2+, CertiNomis ,FR
dn: cn=CertiNomis Classe 2+, o=CertiNomis ,c=FR
certificaterevocationlist;binary:: MIIBJjCBkgIBATANBgkqhki...
certificaterevocationlist;base64:: TulJQkpqQ0JrZ0lCQVRBTkJ...
objectclass: top
objectclass: cRLDistributionPoint
cn: CertiNomis Classe 2+
```





Extensions X.509v3 (6/8)

authorityInfoAccess

- × Indication relative à l'autorité ayant émis le certificat porteur de cette extension, en terme de service de révocation en ligne typiquement (serveur OCSP)

```
Authority Information Access:  
OCSP - URI:http://fd.hsc.fr/
```





Extensions X.509v3 (7/8)

certificatePolicies, policyIdentifier

- * Pointeur vers l'énoncé des pratiques de certification

«[Ce document énonce les] pratiques utilisées par l'IGC dans la gestion des certificats pratiques qui dépendent de la politique de certification mise en oeuvre. Une IGC doit publier cette déclaration afin de décrire les modalités de fonctionnement des service s qu'elle rend.»

FAQ du DCSSI sur les infrastructures à gestion de clef
http://www.scssi.gouv.fr/fr/faq/faq_igc.html

- * Exemple :

```
X509v3 Certificate Policies:  
Policy: 2.16.840.1.113733.1.7.1.1  
CPS: https://www.verisign.com/CPS  
User Notice:  
Organization: VeriSign, Inc.  
Number: 1
```





Extensions X.509v3 (8/8)

X509v3 extensions:

Netscape Cert Type:

SSL CA, S/MIME CA, Object Signing CA

X509v3 CRL Distribution Points:

DirName:/C=US/O=Entrust.net/OU=www.entrust.net/CPS incorp. by ref.
(limits liab.)/OU=(c) 1999 Entrust.net Limited/CN=Entrust.net Secure
Server Certification Authority/CN=CRL1

URI:http://www.entrust.net/CRL/net1.crl

X509v3 Private Key Usage Period:

Not Before: May 25 16:09:40 1999 GMT, Not After: May 25 16:09:40 2019 GMT

X509v3 Key Usage:

Certificate Sign, CRL Sign

X509v3 Authority Key Identifier:

keyid:F0:17:62:13:55:3D:B3:FF:0A:00:6B:FB:50:84:97:F3:ED:62:D0:1A

X509v3 Subject Key Identifier:

F0:17:62:13:55:3D:B3:FF:0A:00:6B:FB:50:84:97:F3:ED:62:D0:1A

X509v3 Basic Constraints:

CA:TRUE

X509v3 extensions:

2.5.29.10:

0.....

commonName:

.)Microsoft Authenticode(tm) Root Authority

2.5.29.1:

0i.....[...*.9..b.S2.R0P1.0...U....US1^M0...U.

..MSFT1200..U...)Microsoft Authenticode(tm) Root Authority...





Faille CryptoAPI (1/2)

« Internet Explorer SSL Vulnerability » (MS02-050)

- × Description du problème
 - × Non prise en compte par Internet Explorer (5, 5.5 et 6) de l'extension v3 `basicConstraints`
 - × Exemple : `X509v3 Basic Constraints : CA:TRUE, pathlen:10`
 - × Signification :
 - × Le certificat est celui d'une autorité de certification : `CA:TRUE`
 - × 10 certificats intermédiaires peuvent constituer la chaîne conduisant aux entités terminales : `pathlen:10`
 - Ces options n'ont de signification que pour le client manipulant le certificat
 - × Absence de l'option critique `basicConstraints` et permissité concernant les `keyUsage`
= possibilité pour la clef privée associée à la clef publique dans le certificat de délivrer un certificat
 - *Un certificat terminal peut se comporter comme une autorité de certification*
 - *Un certificat serveur SSL/TLS valide peut donc certifier une clef publique*
 - Le danger est dans le prolongement de l'itinéraire de certification d'une « autorité de confiance » connue





Faille CryptoAPI (2/2)

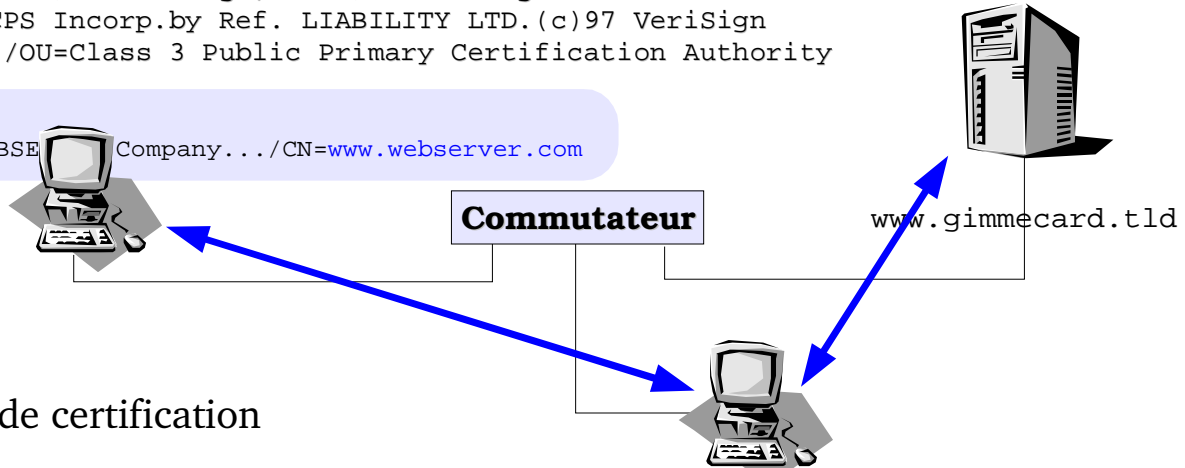
- × Scénario d'exploitation
 - × Attaque de type « *Man-in-the-Middle* »
 - × <http://www.hsc.fr/ressources/presentations/mitm/index.html.fr>

Certificate chain

```
0 s:/C=FR/ST=Hauts de Seine/L=CLICHY/O=WEBSERVER Company.../CN=www.webserver.com
  i:/O=VeriSign Trust Network/OU=VeriSign, Inc./OU=VeriSign International Server CA - Class 3
    /OU=www.verisign.com/CPS Incorp.by Ref. LIABILITY LTD.(c)97 VeriSign
1 s:/O=VeriSign Trust Network/OU=VeriSign, Inc./OU=VeriSign International Server CA - Class 3
  i:/C=US/O=VeriSign, Inc./OU=Class 3 Public Primary Certification Authority
```

s:/CN=www.gimmecard.tld

i:/C=FR/ST=Hauts de Seine/L=CLICHY/O=WEBSERVER Company.../CN=www.webserver.com



- × Prolongation de l'itinéraire de certification
 - × Penser à bien configurer le « relais » présentant le certificat du serveur détourné pour qu'il renvoie à son tour les différents certificats de la chaîne de certification ;-)
- × Extensible à S/MIME !
 - × Faille dans la CryptoAPI (non spécifique à IE)

