Sécurité par compression !

ReSIST 2010

Gilles RICHARD
IRIT www.irit.fr**/BITE** www.bite.ac.uk

Introduction

- Sécurité ..
  - intrusion réseau, attaques (DoS,DDoS etc.)
  - virus, etc...
  - spams
  - code injection (SQL,XSS,CSRF,...)

- Nombreux outils
  - nmap, tcpdump, snort, airodump, etc.
  - anti-virus, spyware, etc.
  - spamassassin, baracuda, outlook/thunderbird
  - fonctions prédéfinies/paramètres de configuration

GRANDE VARIETE -> GRANDE DIFFICULTE;-)

« La Technique »...

analyse « textuelle » des données

- IDS: paquets IP
- spam : header/body
  - recherche de mots clé
  - white/grey/black listes
- DoS or DDoS: paquets IP, log files
- virus : code binaire
  - recherche de signature
  - magic numbers

Pro/cons

PRO :

- ca marche
- human understandable
- paramétrable

CONS :

- facile à contourner
- non auto-adaptatif (spam filter)
- mise à jour nécessaire (mot clé, virus, listes)
- inefficace contre NOUVEAUTE : Zero-Day attack

SI ON ESSAYAIT AUTRE CHOSE;-)

Une autre idée !

- dans tous les cas,
  - capturer des données
  - considérer leur sémantique/**contenu informatif**
  - comparer ce “contenu” à qque chose de stable
  - conclure et agir !
- on a besoin de
  - notion claire de contenu informatif (température ?)
  - notion quantifiable numériquement
  - capacité à le mesurer ou estimer au moins
  - et ensuite comparer...

SOLUTION... bien avant Ipad ;-)

Kolmogorov

- Andrei Kolmogorov (1903-1987)
- Théorie des probabilités
- Théorie algorithmique de l'information 1964
- Complexité de Kolmogorov
  - Information
  - Hasard↗ Information ↗
  - Hasard↗ Apprenabilité ↘
  - Hasard↗ Compressibilité ↘

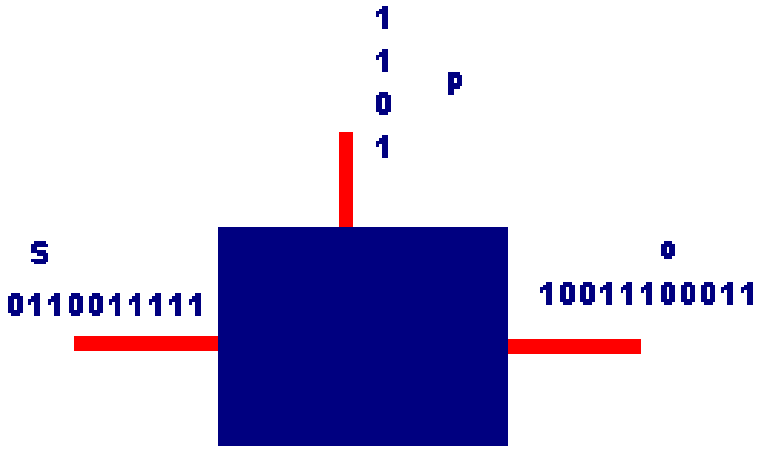
Machine de Turing

C'est quoi ?

Simple modèle de PC

3 rubans seulement = calculateur T

- Input y
- Program p
- Output x = T(p,y)



K(x)

Définition:

- Choisir un (universel) calculateur **T**
- **K(x)** = longueur du plus court programme **p** tel que **T(p, Ø) = x**

- K(x) < |x| + c (expliquer)
- ne dépend pas vraiment de T (i.e. du langage)
- Constante « universelle” associée à une chaine x

Des exemples

- x = 1 milliard de fois 010101....
  - `for i = 1 to 10^9 {print 0 ; print 1} ;`
  - K(x) faible par rapport à |x|
- x = 01101000100111111101100001111 ...
  - simplement réecrire : `print(x)`
  - K(x) de l'ordre de |x|
- x = PI (10^9 premiers digits)
  - 100lignes de C
  - K(x) = 100*10*10 = 10000
- en pratique: mesure du hasard/désordre !
- lien entre **prédire/apprendre/générer**

Estimer K ?

- K(x) nombre idéal non calculable
- Calculateur ...compresseur
  - une chaine x
  - un algo de compression sans perte C : C(x)
  - calculateur T = algo de décompression C^{-1}

T(C(x),Ø) = x

- |C(x)| approxime K(x) : K(x) < |C(x)|
- zip, compress (LZ), bzip2, 7zip(LZMA)

K(x)<7zip(x)<bzip2(x)<compress(x)<zip(x)

A quoi ca sert ?

Initiateur: Stephen Bush (General Electric - 2000)

http://www.crd.ge.com/~bushsf/

- K = température d'un système
  - bonne santé = variations raisonnables
  - problème si K trop bas ou trop haut
- en pratique :
  - capture de paquets IP (tcpdump) -> fichier binaire (taille constante)
  - transformation code ASCII -> fichier ascii
  - compression du fichier
  - estimation de K

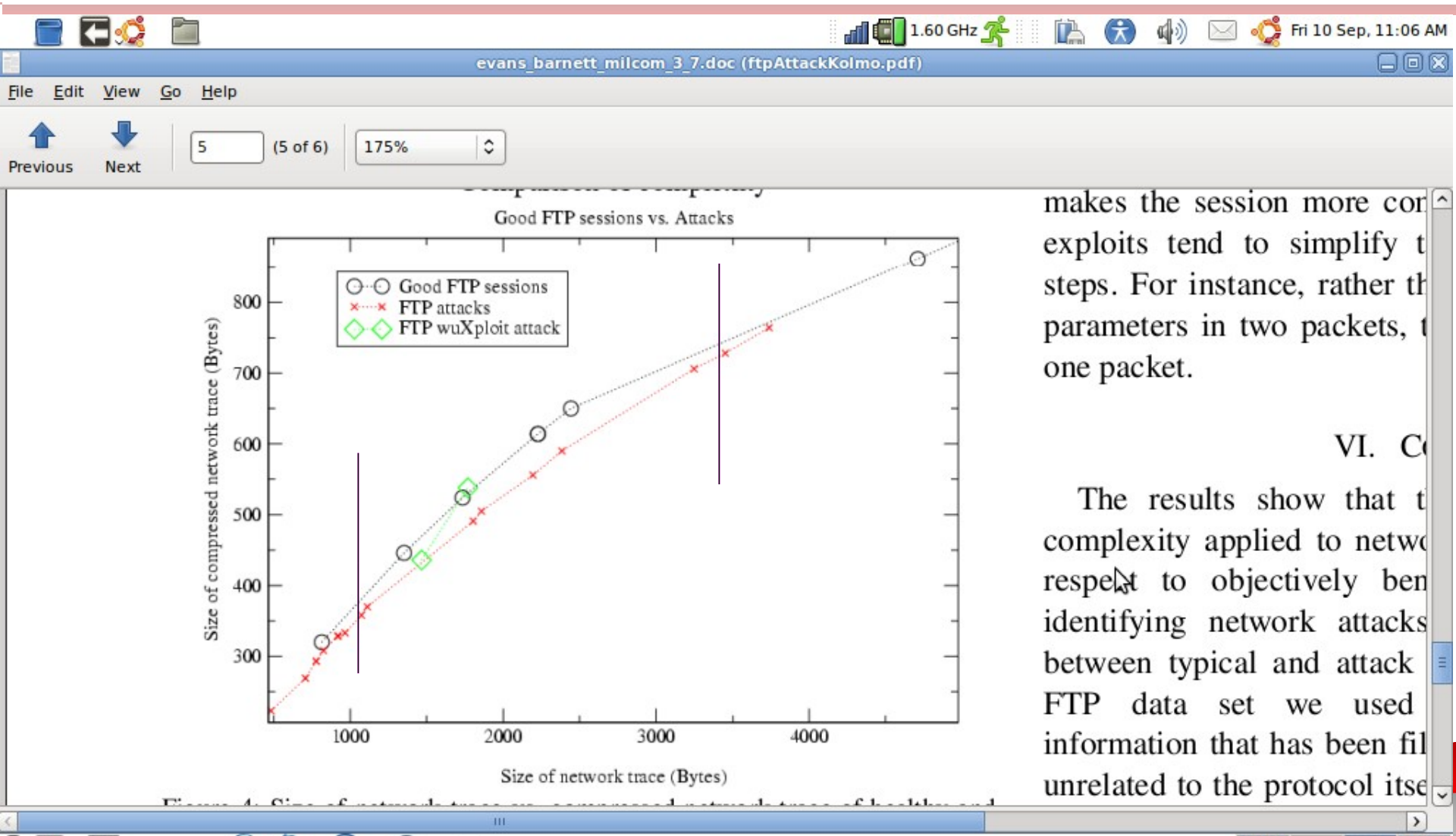
if K < min OR max <K then ALARM

Détection d'attaques FTP

Auteurs : Scott Evans & Bruce Warnett (GE research center) - 2002

- Expériences seulement (pas de produit commercial)
  - serveurs ftp : proftpd,wu-ftdpd,wftppro
  - client ftp : expect, ncftp ou netscape
  - capture : tcpdump sur firewall
  - compression : compress (Unix) (utilise LZ)
  - clients et serveurs Unix
  - sessions “normales”
- Génération d'attaques FTP
  - avec des outils commerciaux
  - 1000 attaques (DoS,buffer overflow,etc.)
- Etudier courbe de K **normale** versus **attaque**

Les courbes


1.60 GHz
Fri 10 Sep, 11:06 AM
evans_barnett_milcom_3_7.doc (ftpAttackKolmo.pdf)
File Edit View Go Help
Previous
Next
5
(5 of 6)
175%
Good FTP sessions vs. Attacks
Good FTP sessions
FTP attacks
FTP wuXploit attack
Size of compressed network trace (Bytes)
Size of network trace (Bytes)
300
400
500
600
700
800
1000
2000
3000
4000
makes the session more cor
exploits tend to simplify t
steps. For instance, rather th
parameters in two packets, t
one packet.
VI. C
The results show that t
complexity applied to netwo
respect to objectively ben
identifying network attacks
between typical and attack
FTP data set we used
information that has been fil
unrelated to the protocol itse

Résultats FTP

- Précision: 95%
- FPR: 0.2% (fausses alarmes)
- Difficultés :
  - déterminer les seuils
  - quantité de données à compresser : 2KB, 3KB ?
  - compression temps réel : balance à trouver
    - précision
    - temps d'exécution
- Courbes à étudier pour http, imap,ssh, etc...
- Vision locale (1 machine)

Scanning

Auteurs : Eiland/Liebrock (University of New Mexico) - 2006

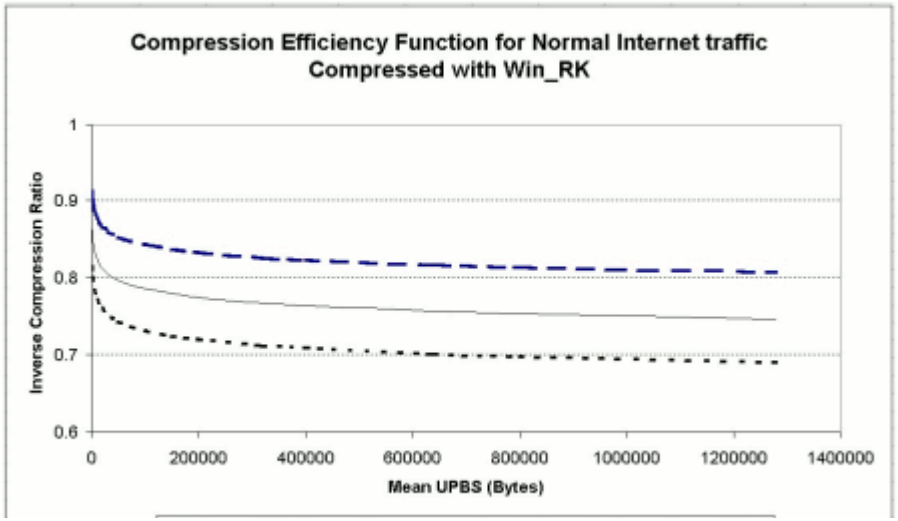
- Même idée pour attaque nmap ()
  - nmap -sS ,
  - nmap -sX, ...
- Précède en général une attaque réelle
- K ne donne rien ! ... donc...
- Normaliser entre [0,1] avec Inverse Compression Ratio (ICR)

$$ICR(s) = K(s)/|s|$$

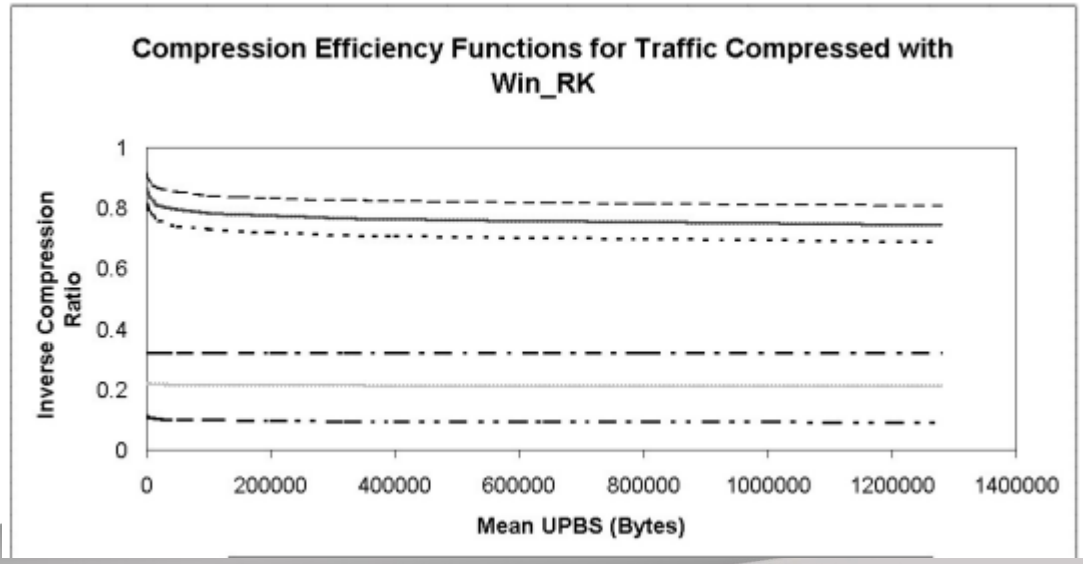
- Etudier la courbe ICR en fonction de |s|
- Connaitre la “bonne” quantité de données à compresser
- Compresseur : WinRK

Résultats

- la courbe normale

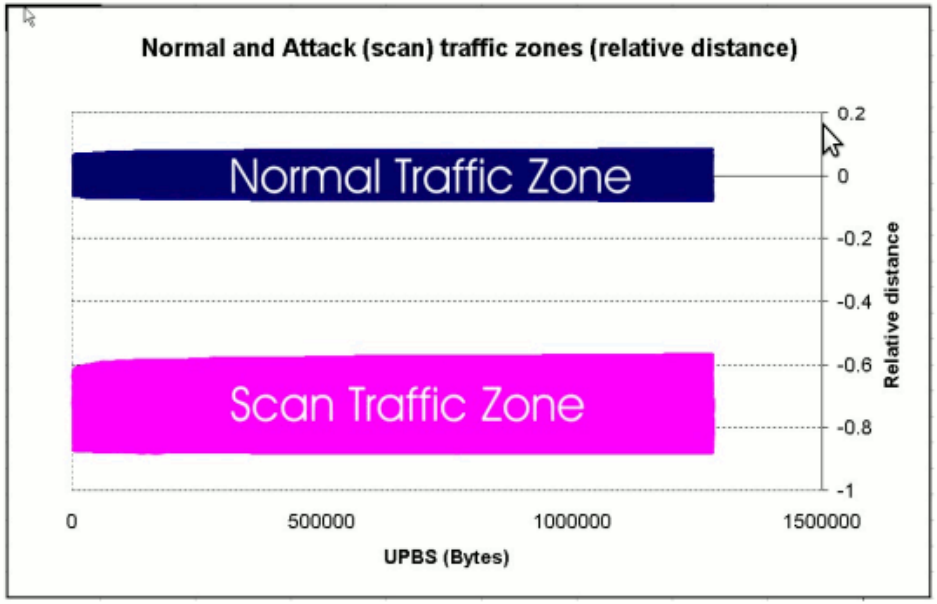


- les 2 courbes



La bande normale

- plus de données à compresser (optimal = 320KB)
- séparation état normal/anormal
- précision: 95% FPR 0.3% (standard : 92%)



Autre idée

Auteurs : R. Cilibrasi (CWI - Amsterdam) - S. Bush 2002

- Détermination du type des données
  - ASCII,
  - audio,
  - video,
  - pictures
  - exe,
  - doc

- En pratique :
  - capturer des flux de donnèes
  - compresser
  - donner une prèdiction du type de donnèes encapsulèes

Performances

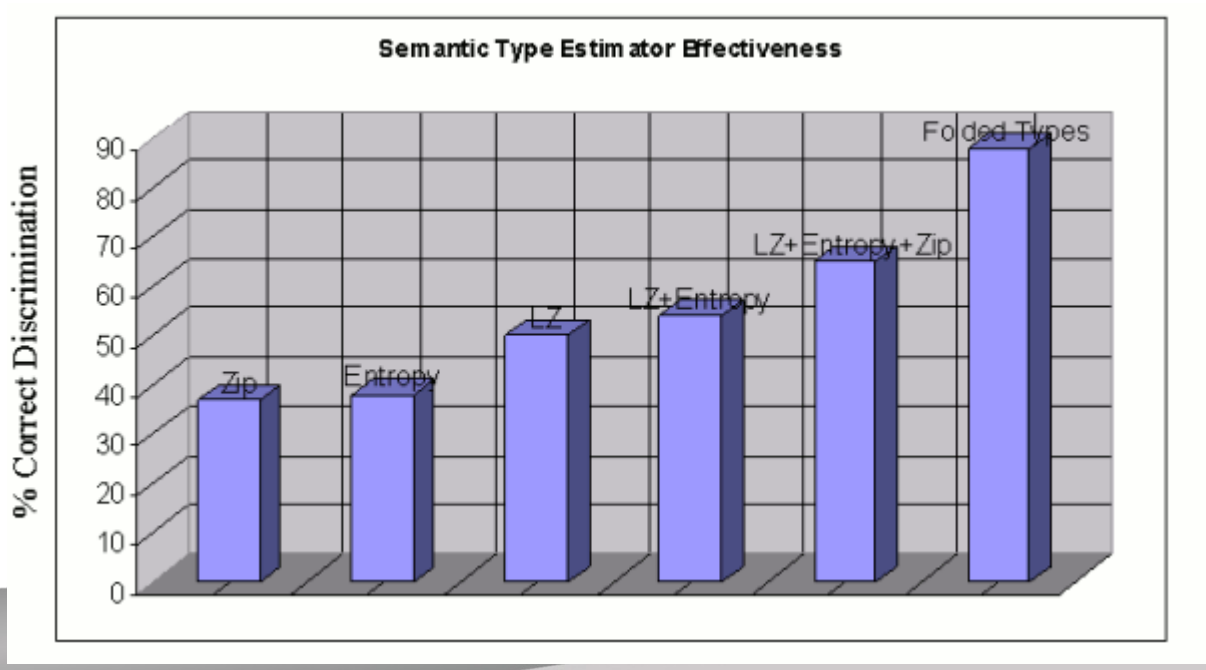
Too close to call, perhaps they shoul
collapsed into the same semantic ty

- Discrimination

Squared Distance Between Groups

	Audio	Doc	Exe	Pic	Txt	Vid
Audio	0.0000	10.5632	0.0453	3.5858	12.1523	4.6602
Doc	10.5632	0.0000	9.8092	26.1292	0.2660	29.1583
Exe	0.0453	9.8092	0.0000	3.9334	11.5128	5.3209
Pic	3.5858	26.1292	3.9334	0.0000	28.8806	0.5649
Txt	12.1523	0.2660	11.5128	28.8806	0.0000	31.3792
Vid	4.6602	29.1583	5.3209	0.5649	31.3792	0.0000

- Précision en fonction des types de données



Discrimination

- Avantages :
  - pas de recherche de signature
  - pas d'analyse de header
  - aucune info a priori
  - détection executables cachés (virus?)
- Problèmes :
  - Distinguer les types proches en terme de compression
  - Trouver la quantité optimale de données à compresser

Autres applications/méthodes

- Applications :
  - DDoS détection (Kulkarni,Bush,Evans 2001)
  - Vulnerability assessment via compression
  - Détection de virus (Goel&Bush 2003)
  - autres probablement;-)
- Méthodes :
  - distance (différentiel) à état normal (expliquer)
  - data mining techniques

Peut on faire mieux ?

- Distance : outil puissant pour classifier
- avec K : Distance informationnelle (Bennett)

- a: K(a) ... = K(a - b) + K(a ∩ b))
- b: K(b) ... = K(b - a) + K(a ∩ b))
- a.b: K(a.b) = K(a - b) + K(b - a) + K(a ∩ b))
- m(a, b) = K(a) + K(b) - K(a.b) : mesure du contenu commun en information

d(a, b) = 1 - m(a,b) / max(K(a), K(b)) in [0,1]

- Propriétés d'une distance (expliquer)
- d(a,b)=0 ssi a=b
- d(a,b)=1 si aucun contenu commun

Spam filtering

- k-NN algorithme
- principe :

  “qui se ressemble s'assemble”
- Plusieurs idées
  - base de spam/ham
  - email e -> table des distances spam/ham
  - vote majoritaire
  - test sur Spamassassin data sets
  - test set de 1000 emails

Résultats avec k-NN

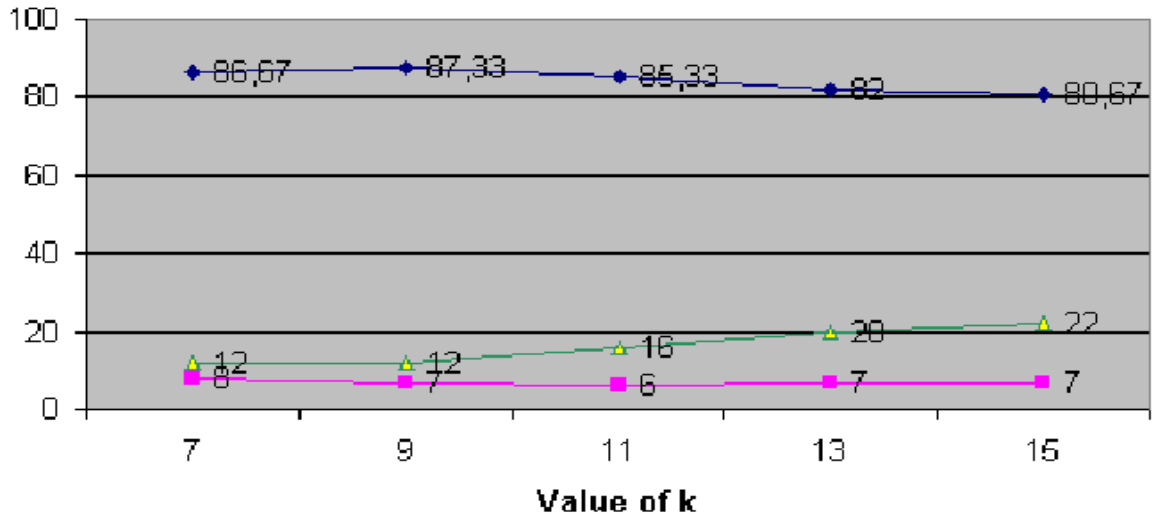
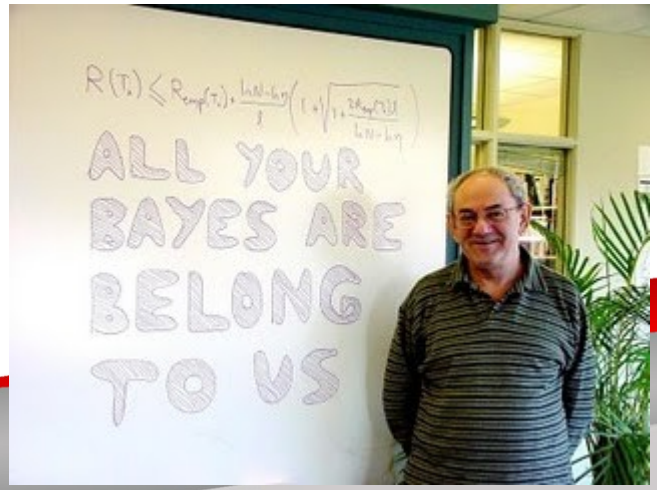


Figure 1: 50 spam/50 ham witness e-mails.

Spam filtering... plus fort !

- SVM algorithme (V. Vapnik)
- Séparation linéaire versus noyau
- Très performant en général
- Transformer emails en vecteurs réels
  - choix de 10 typical spam/10 typical ham
  - calcul des distances pour m donné -> vecteur de 20 composantes



Résultats SVM

train50	8	10	12	16	20
Accuracy	94.60	95.60	95.40	94.60	95.00
FPR	1.60	2.80	4.00	5.60	5.20
TPR	90.80	94.00	94.80	94.80	95.20
train100	8	10	12	16	20
Accuracy	94.60	96.20	95.40	94.80	94.80
FPR	1.60	2.40	2.40	1.60	1.20
TPR	90.80	94.80	93.20	91.20	90.80
train150	8	10	12	16	20
Accuracy	94.80	94.60	94.60	96.40	96.20
FPR	1.20	2.40	2.40	2.40	2.40
TPR	90.80	91.60	91.60	95.20	94.80
train200	8	10	12	16	20
Accuracy	93.80	93.40	93.60	93.80	93.40
FPR	1.60	1.60	2.00	1.60	1.60
TPR	89.20	88.40	89.20	89.20	88.40

Spam filtering... pro/cons !

- auto-adaptatif avec le training set
- incontournable par spammer
- Pb :
  - les images... déja compressées !
  - les SPIT ;-)

Finally !

- Plus d'analyse (utile pour MANET)
- Plus de mise a jour
- Force “brute” de la compression
- Panacée universelle ?? NON...
  - SQL injection, XSS ??
  - couplage
  - implémentation performante
  - etc...

Never ending story !

Kolmogorov to Solomonov...

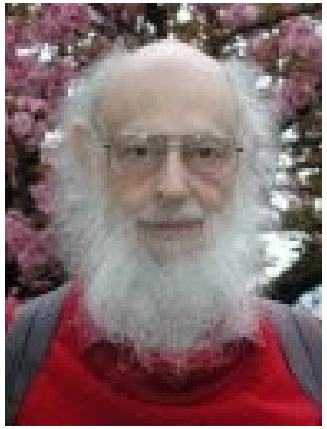
Solomonov distribution (universal)

$$p(x) = 2^{-K(x)}$$

Complex systems are unlikely ;-)

Main idea:

Instead of ... **K 2 P**



Work the reverse way... **P 2 K**

$$K(x) = -\log(p(x))$$