



Netzob : un outil pour la rétro-conception de protocoles de communication

Georges Bossert ^{1 2}, Frédéric Guihéry ¹, Guillaume Hiet ²

¹ AMOSSYS - Rennes, France

² Research team CIDre, Supélec

6 juin 2012



Georges Bossert



Doctorant
AMOSSYS / Supélec

Frédéric Guihéry



Ingénieur Sécurité
AMOSSYS

Guillaume Hiet



Enseignant chercheur
Supélec - CIDre

- **AMOSSYS** : Conseil et Expertise en Sécurité des Technologies de l'Information.
- **SUPELEC - CIDre** : Groupe de recherche focalisé sur la sécurité des systèmes d'informations distribués.



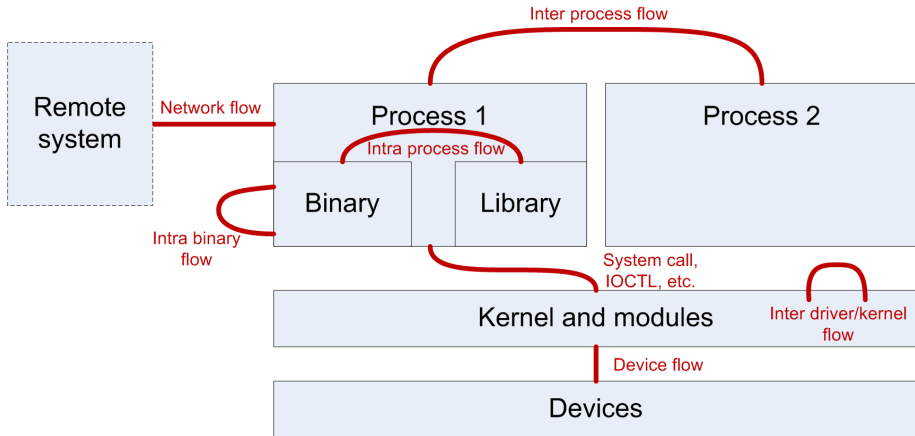
Plan

- 1 Le contexte
- 2 Notre modèle d'un protocole
- 3 L'inférence du modèle
- 4 Présentation de l'outil Netzob



Le contexte

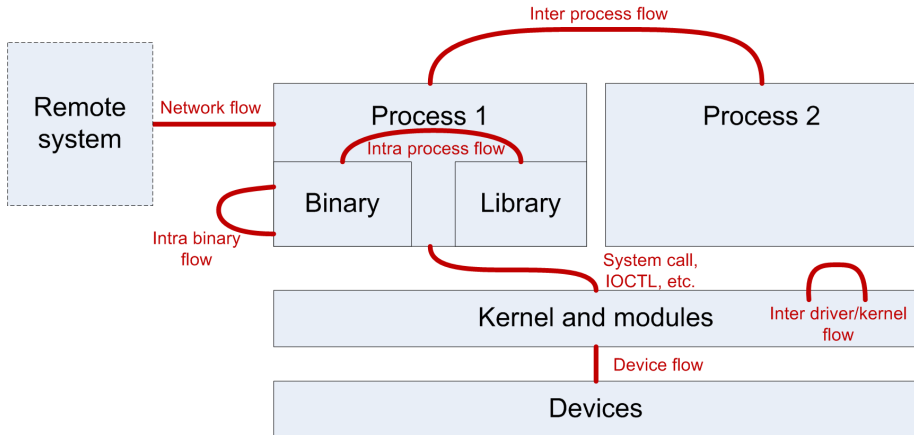
Omniprésence des protocoles de communication





Le contexte

Omniprésence des protocoles de communication



Malheureusement, les spécifications ne sont pas toujours disponibles
(protocoles propriétaires ou non documentés)



Pour **évaluer** la robustesse des implémentations

=> Fuzzer l'API de contrôle d'une centrifugeuse.

Pour **générer** un trafic réaliste et contrôlable.

=> Simuler la présence d'un botnet.

Mais également...

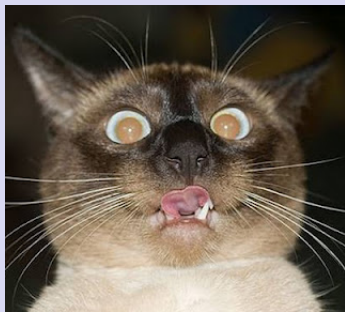
- Pour **analyser** le trafic et identifier d'éventuelles fuites de données.
- Pour **développer** une version libre d'une implémentation propriétaire.
- Pour **valider** l'implémentation d'un protocole vis-à-vis d'une spécification.



L'essentiel du temps de rétro-analyse d'un protocole ressemble à ça :



L'essentiel du temps de rétro-analyse d'un protocole ressemble à ça :

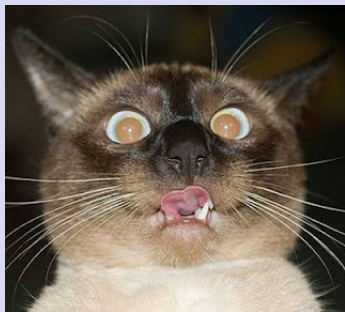


```
4f945a3...9e0bb1c2722165b349774c607deb34efb5ef1878aea36bc6ae7d9be  
89546ef16a59c959f592ae5297385abf71c297a...  
a20785cd3b10e2544290bec164aaad64f84f6868d4495bd50d332340276904ef00629808ff7fa2f2  
70c5c3c66a3edde50b8347fae58ed2bcb2d87401c86c4e1600e9ff...  
09416f06d6c3b24a7fadad6c2aa7f0c3da7caf5d2311dd6d0e08640c290b366bd13400cbff19509b  
1200ed4ab3c40859076878a9391c60147009ff13ee10d193d256eb53c1c8728ac  
p42f37b5e5853490cbff1188d51c08ce8e78e...B64(Y)  
4cc894ea8bd549308eba364100bef776f888bec5ed91448b788cd1d382c5a95a0411acd885927  
f5f4b9eb3b781ff10656f2b734e9f3340f75ba9a62db0f0970adf62feeda7b887e0d66311e46a9f  
479a2ca40c82a51a948cb4346b89d17888d9e057b521ccb5a759888bb568d2bbe4a406e41fc3028  
32af01d017e2a8ed00c7501b7599c0e30...  
bbbb0df3b088703f8e9842080df1935004c09979744ff393639023b62190ceca64a4eede5d757ac  
bfdd70b1004efbcb9859183ba3608f4749400ab2401ae22723...  
16a58d51d5ba103774459ff8e1e0ff8c6afad206c2014cca24f171475905ac3...  
11408fde92c00d3fff00ca9c3c38...  
0a691e8a3fe3c9423d99c427ba81200edff842d1c115254e573d4a2730927d3e4d03e69300ccffdd  
4313000cfff872cc766b40aa3e01490ebcea54cf3def3fc4f91edb5674840d9446204b566716d3ad9  
4178d0ade5039d0f1344662dfbf1043c642d6d34581337eedb73628703d35249911a42691ae038d  
a78b3fc0c831f8f57bbb4f0dc216bb67c354149ad639add4bd026cfa0110cfeaf02101800e77fc  
ebb3998a8ed59daf6ff6d0c01663c0f305c1d39110898a0b...  
ff46e530e92b21bf1fe07ced1fe942f0a177a1ba802ab...  
e5e8d85d6d4e222dd432f00d0ff3127e2003add173565f...  
79ac8ad48b3929e603c2731480321869fec0df02d45c3a95d3f036c356c984ee031992cd253e50  
d668979894b...  
73855f232d7f492ead9f9f0bdc27f3...  
333f3e210446d6b43674100bef256f0...  
01fab57c515469b7af29e972cc4800b...  
41800e7f1487180f123ae0ba2b0f9808555c64495106b0f6370b5adclb00e4ffa3293f10c8743f9a  
aenc190ab07cdca9e3c8a566176cf1bd402913cceb221b61d61d97321e19e3368dfe7706728d  
888a9c9f56f789796a0bd34c0a00f5fcaf0c85ef8f1118f604c08007ffccfeacacbd8a2621800  
9e8b0c38aad085d41100eef233a767ce75b081493af803e57a76683e31400ebff8baed3dfedc446
```

- Processus complexe et laborieux
- Processus essentiellement manuel, voire visuel
- Communauté dépourvue d'outils d'analyse



L'essentiel du temps de rétro-analyse d'un protocole ressemble à ça :



4f945a3...9e0bb1c2722165b3497f4c607deb34efb5ef1878aea36bc6ae7d9be
89546ef16a59c959f592ae5297385abf71c297a...
a20785c...bec164aad64f84f6868d4495bd50d332340276904ef00629808f7ffa2f2
70c5c3c66a3edde50b8347f5ae58ed2bc2d287401c86c4e1600e9ff...
09416f06d6c3b24a7adad6c2aa7f0c3da7caf5d2311dd6d0e08640c290b366bd13400cbff19509b
c1200ebd4ab3c40859076878af9391c60147009ff13ee10d193d256eb53c1c8728ac
p42f31f0b0e583490cbff1188d51ce08ce878e...B64(Y)
4cc8b94ea8bd54938eba364100bef776f888bec5ed91448b788cd1d382c5a95a0411acd885927
f5f4b9eb3b781ff10656f2b734e9f3340f75ba9a62db0f0970ad6f2feeda7b887e0d66311e46a9f
479a2ca40c82a51a948cb4346b891d7888d9e057b521ccb5a759888bb568d2bbe4a406e41fc3028
32af01d017e2a8ed00c7501b7599c0e3...
bbbb0df3b088703f8e9842080df93500c0997f744ff393639023b62190ceccaf4a4eed5d75fac
bf7db1004ef7fbc9859183a6a98f4749400ab2401ae22723...
16a58d51d5ba103770469ff8e1e0ff8c6afad206c2014cca24f171475905ac3...
11408bfade92c00d3fff50ca9c9c38...
0a691e8a3fe3c9423d99c427ba81200edff842d1c115254e573d4a2730927d3e4d03e69300ccffdd
4313000cfff872cc766b40aa3e01490ebcea54cf3def3fc4f91edb5674840d9446204b566716d3ad9
4178d0ade5039d0f1344662dfbf1043c642d6d34581337eedb736287f3d3d35249911a42691ae038d
a78b3fc0c8318f57bbb4f0dc216bb67c354149ad639add4bd026cefa0110cfa82101800e7ffcc
ebb3996a8ed59daf6ff6dc01663c0f305c1d39110898a0b...
ff46e530e92b21bf1fe07ced1fe942f0a177a1ba802ab59077f80ec3e
e5e8d85d6dae222dd432f00d0ff3127e2003add173565f...
79ac8bad48b3929e603c2731480321869fec0df02d45c3a95d3f036c356c984ee031992cd253e50
d668979894b...
7385557232d7f492ead9f9f0dc2d7f3...
333f3e210446d6b43674100bef256f0...
01fab57c5f5469b7af29e972cc4800b...
41800ef7f487180f123ae6ab20f09805556c4495106b0f6370b5adclb00e4f3a3293f10c8743f9a
a6ec190ab07cd6ca9e38a566176cbf1bd402913cceb221b61f61d97321e1a93368d6f7706728d
888a9c95779976ab0bd3ac00a00f5f7cfa0c05ef8f1118f604c0800f7ffccfeacacbd8a02621800
9e8b0c038aad085d41100eef233a767ce75b081493af803e57a76683e31400ebff8baed3dfedc446

- Processus complexe et laborieux
- Processus essentiellement manuel, voire visuel
- Communauté dépourvue d'outils d'analyse

Ce constat, partagé, a mené à la création du projet Netzob



Protocole informatique : spécification

- Vocabulaire : liste des messages et leur format.
- Grammaire : règles de procédure permettant d'assurer la cohérence des messages échangés.



Protocole informatique : spécification

- Vocabulaire : liste des messages et leur format.
- Grammaire : règles de procédure permettant d'assurer la cohérence des messages échangés.

Rétro-conception

- Retrouver la spécification de protocoles lorsqu'elle est inconnue
 - Par **inférence** (passive ou active) ou par analyse (manuelle)
 - A partir des **messages échangés** ou de l'exécutable



Plan

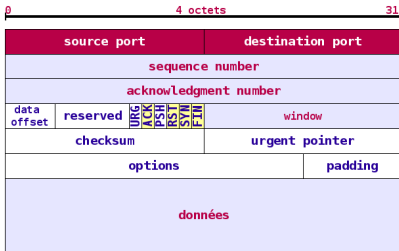
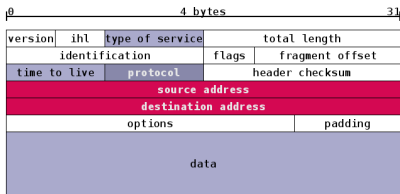
- 1 Le contexte
- 2 Notre modèle d'un protocole
- 3 L'inférence du modèle
- 4 Présentation de l'outil Netzob



Le modèle du vocabulaire

Identification des besoins

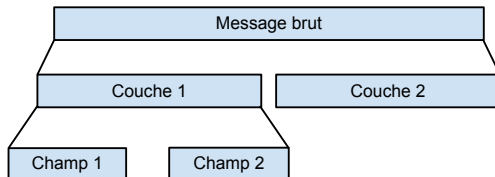
Exemples que l'on souhaite vouloir modéliser

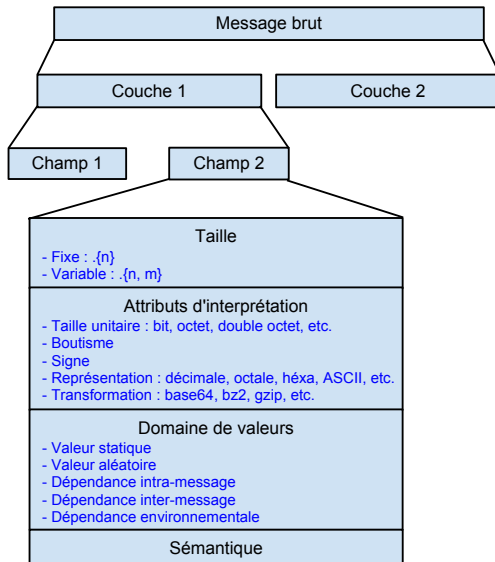




Le modèle du vocabulaire

Réponse aux besoins







#1 : Traduire la relation entre un symbole en entrée et un symbole en sortie

- Répondre « *attack successful* » lorsque l'on reçoit « *attack* ».



#1 : Traduire la relation entre un symbole en entrée et un symbole en sortie

- Répondre « *attack successful* » lorsque l'on reçoit « *attack* ».

#2 : Définir plusieurs symboles de sortie pour le même symbole d'entrée

- Répondre « *attack successful* » ou « *attack failed* ».



#1 : Traduire la relation entre un symbole en entrée et un symbole en sortie

- Répondre « *attack successful* » lorsque l'on reçoit « *attack* ».

#2 : Définir plusieurs symboles de sortie pour le même symbole d'entrée

- Répondre « *attack successful* » ou « *attack failed* ».

#3 : Associer une probabilité d'émission pour chaque symbole de sortie

- « *attack successful* » (90%) ou « *attack failed* » (10%).



#1 : Traduire la relation entre un symbole en entrée et un symbole en sortie

- Répondre « *attack successful* » lorsque l'on reçoit « *attack* ».

#2 : Définir plusieurs symboles de sortie pour le même symbole d'entrée

- Répondre « *attack successful* » ou « *attack failed* ».

#3 : Associer une probabilité d'émission pour chaque symbole de sortie

- « *attack successful* » (90%) ou « *attack failed* » (10%).

#4 : Prendre en compte le temps de réponse

- « *attack successful* » plus rapide que « *attack failed* ».



Machine de Mealy Stochastique à Transitions Déterministes

$$MMSTD = \langle S, X, Y, T, q_0 \rangle$$

q_0 État initial

S Ensemble des états

X Alphabet des symboles d'entrées

Y Alphabet des symboles de sorties

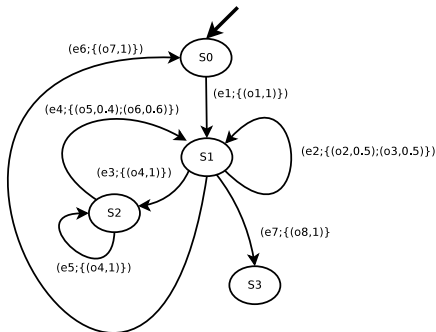
T Ensemble de matrices de transitions

- $|T| = |X| \times |Y|$
- $T = \{A(x, y)\}, a_{i,j}(x, y) = (p(s_j, y | s_i, x), t_{i,j}(x, y))$
- $\forall x \in X, \forall s_i, s_j \in S \quad p(s_j | s_i, x) = \sum_{y \in Y} p(s_j, y | s_i, x) \in \{0, 1\}$



Le modèle de la grammaire

Machine de Mealy Stochastique à Transitions Déterministes



Alphabets :

Entrées

e1=".login 123"
e2=".info"
e3=".ddos"
e4=\$IP
e5=!\$IP
e6=".logout"
e7=".disconnect"

Sorties

o1="Welcome master"
o2="Windows XP"
o3="Linux"
o4="IP of target ?"
o5="Attack successfull"
o6="Attack failed"
o7="Goodbye"
o8="Disconnected"

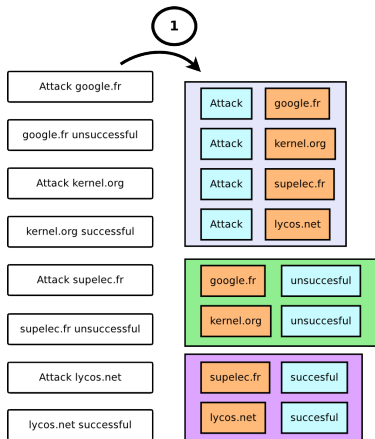
Pour résumer :

- Automate symbolique
- **Transitions déterministes** mais symboles de sorties indéterministes
- Prise en compte du **temps de réaction**



Plan

- 1 Le contexte
- 2 Notre modèle d'un protocole
- 3 L'inférence du modèle**
- 4 Présentation de l'outil Netzob



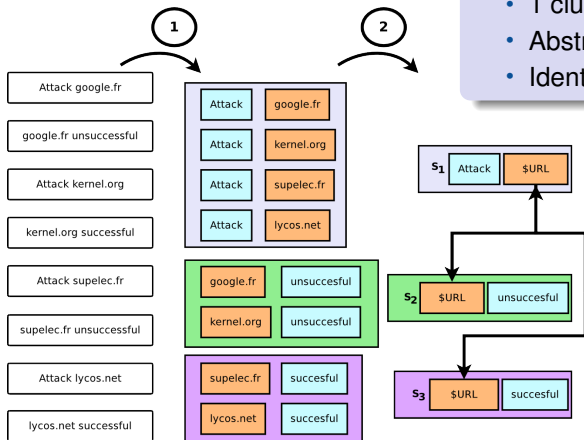
#1 : Découpage et regroupement

- Découpage en champs
- Regroupement des messages par similarité
- Approche semi-automatique



#2 : Abstraction en symboles

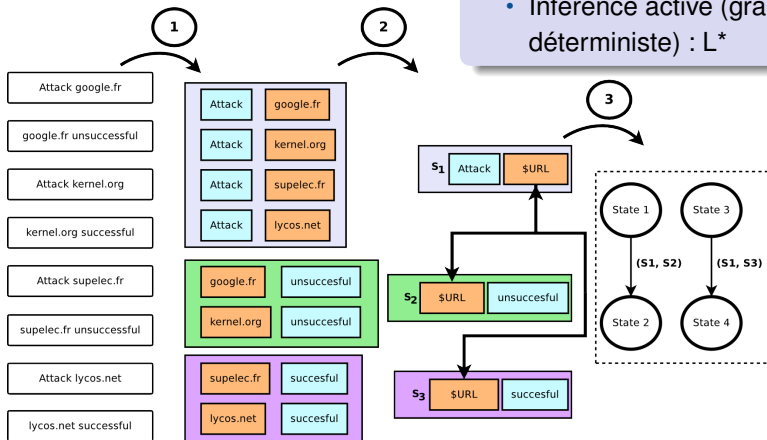
- 1 cluster = 1 symbole
- Abstraction des champs
- Identification des dépendances





#3 : Inférence du graphe de transition

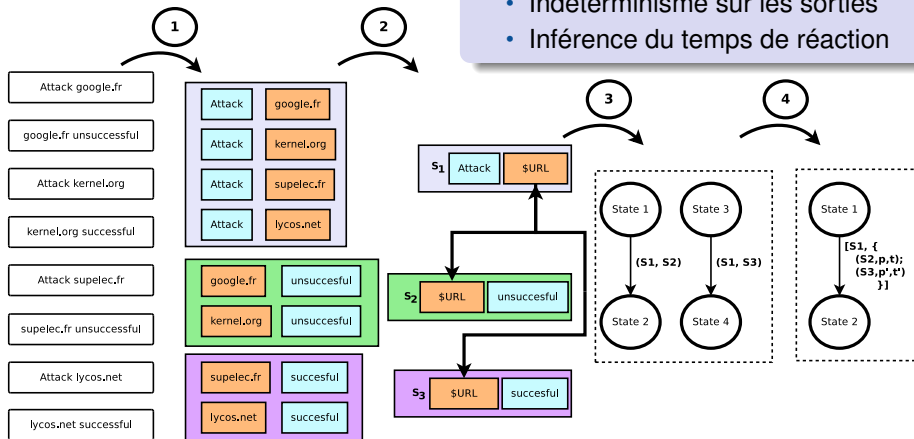
- Inférence active (graphe déterministe) : L^*





#4 : Généralisation de l'automate

- Indéterminisme sur les sorties
- Inférence du temps de réaction





Le découpage en champs

- Partitionnement par délimiteur
- Partitionnement par variation

0	2	010600	00003d1e	0000000000000000	c0a8000a00000000	0000000000	0b8201fc42	00000000
0	2	010600	00003d1d	0000000000000000	c0a8000ac0a80001	0000000000	0b8201fc42	00000000
0	1	010600	00003d1e	0000000000000000	0000000000000000	0000000000	0b8201fc42	00000000
0	1	010600	fe089c15	0000000000000000	0000000000000000	0000000000	50ba1247cb	00000000
0	1	010600	9a5a2277	0000000000000000	0000000000000000	0000000000	1cc07e38c3	00000000
0	2	010600	fe089c15	0000000000000000	0a1414140a141404	0000000000	50ba1247cb	00000000
0	2	010600	fe089c15	0000000000000000	0a1414140a141404	0000000000	50ba1247cb	00000000
0	2	010600	9a5a2277	0000000000000000	c0a8000e00000000	0000000000	1cc07e38c3	00000000
0	2	010600	33dca406	0000000000000000	c0a8000e00000000	0000000000	1cc07e38c3	00000000

Champs dynamiques à taille fixe

- Alignement de séquence (Needleman-Wunsch)

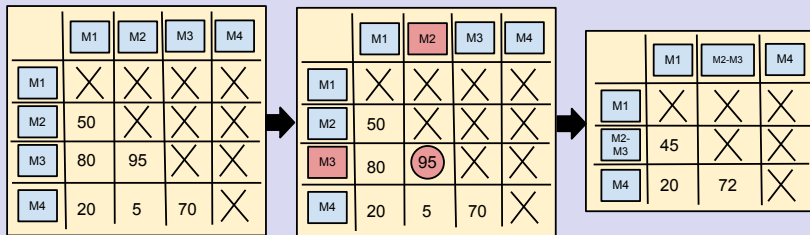
150 Opening BINARY mode data connection for	tb.php	(5292	bytes)..
150 Opening BINARY mode data connection for	rss.php	(3965	bytes)..
150 Opening BINARY mode data connection for	coords.txt	(365	bytes)..
150 Opening BINARY mode data connection for	access_marsToMai.log	(1392891	bytes)..

Champs dynamiques de taille variable



Regroupement par clustering UPGMA

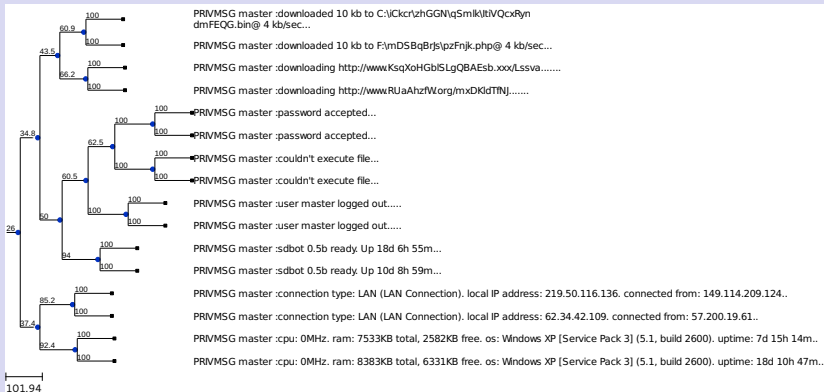
- Calcul de similarité en sortie de Needleman-Wunsh
- Remplissage d'une matrice de similarité
- Fusion des messages les plus similaires
- Puis on itère...





Regroupement par clustering UPGMA

- Représentable sous la forme d'un dendrogramme
- Possibilité de choisir un seuil de regroupement





Inférence du graphe de transition

- **Principe** : soumettre des séquences de **symboles** à un « professeur »
- Transmission des messages vers/depuis le binaire confiné
- Variante de l'algorithme L^* (Angluin).
- Ré-initialisation entre chaque soumission (virtualisation)

Généralisation du modèle

- Ajout de l'indétermination en sortie :
 - rejeu des traces dans l'automate inféré
 - estimation des probabilités des symboles de sorties
- Ajout du temps de réaction
 - Mesure du temps entre un symbole d'entrée et de sortie.
 - Utilisation d'une loi normale.



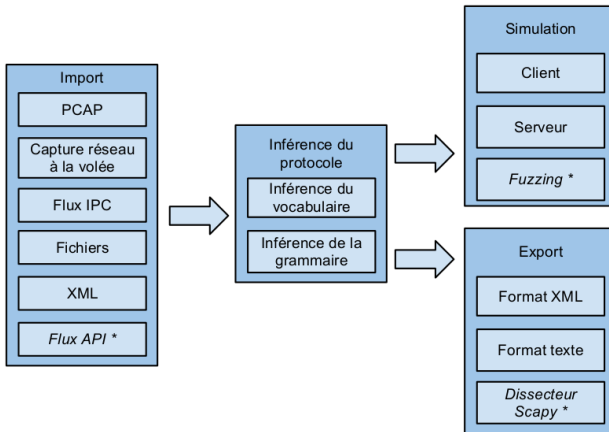
Plan

- 1 Le contexte
- 2 Notre modèle d'un protocole
- 3 L'inférence du modèle
- 4 **Présentation de l'outil Netzob**



Présentation de l'outil Netzob

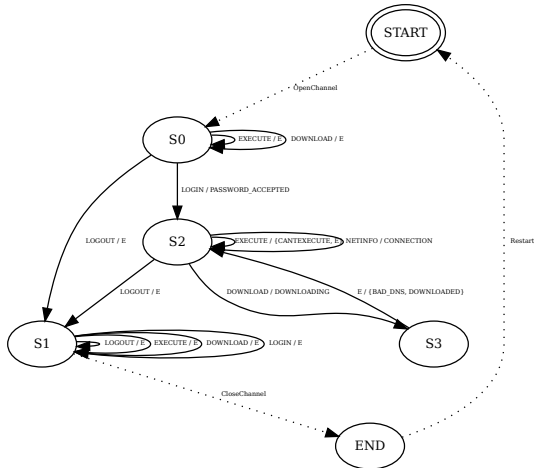
L'architecture fonctionnelle





[Démo]

The screenshot shows the Wireshark interface with a packet list on the left, a packet details pane in the center, and a context menu open on the right. The packet list shows a packet with IP source 192.168.0.14 and IP target 192.168.0.254. The packet details pane shows the selected packet's structure: Ethernet II, Internet Protocol Version 4, and User Datagram Protocol. The context menu is open, showing options like 'Move to ...', 'Edit field', 'Field visualization', 'Concatenate field', 'Split field', 'Partitioning', 'Field's domain of definition', 'Configure variation of field', 'Copy to clipboard', 'Message properties', and 'Delete message'. The 'Field visualization' option is highlighted.







- Développé en Python et en C
- Sous licence GPLv3
- Actuellement 7 contributeurs (AMOSSYS, Supélec, Bull)
- Disponibilité
 - Via dépôt git ou tar.gz
 - Paquet Debian (À la recherche d'un sponsor)
 - Paquet Gentoo et Windows (en cours)

Network protocol modelization by reverse engineering
still in dev.

NETZOB

LATEST AND GREATEST (VERSION 0.3.1)
[changelog](#) | [past releases](#) |  

HOME ABOUT DOWNLOAD DOCUMENTATION CONTACT


WHAT IS NETZOB ?

NETZOB SIMPLIFIES THE WORK FOR SECURITY AUDITORS by providing a complete framework for the REVERSE ENGINEERING OF COMMUNICATION PROTOCOLS. It handles different types of protocols : text protocols (like HTTP and IRC), fixed fields protocols (like IP and TCP) and variable fields protocols (like ASN.1 based formats).

Netzob is therefore suitable for REVERSING NETWORK PROTOCOLS, STRUCTURED FILES AND SYSTEM AND PROCESS FLOWS (IPC and communication with drivers). Netzob is provided with modules dedicated to capture data in multiple contexts (network, file, process and kernel data acquisition).

Besides being intrinsically adapted for classical reverse engineering of protocols, Netzob is capable of IDENTIFYING POTENTIAL VULNERABLE DATA FIELDS, like size fields and their associated payloads.

Furthermore, Netzob integrates a STOCHASTIC AND STATEFULL MODEL to represent any communication protocol through an XML declaration. This model declaration is then loaded in a dedicated component of Netzob, its network simulator. Therefore, it becomes easy to SIMULATE MULTIPLE ACTORS (servers and clients) which communicates according to the inferred protocol for advanced fuzzing processes or active inferring process.





Bilan

- Domaine assez actif au niveau académique ces dernières années
- Mais quasiment aucune retombée dans les outils publics
- Netzob vise à combler ce manque
 - Support de travaux académiques
 - Utilisable dans un **contexte opérationnel** (audit, évaluation, développement, ...)

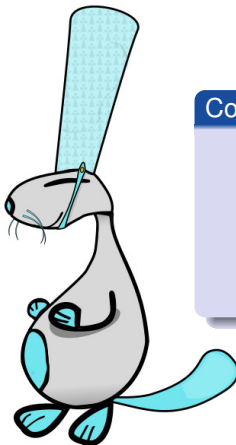
Travaux futurs

- Nouveaux capteurs (API, noyau, périphériques)
- Gestions des sessions
- Gestion d'autres formats d'encodages (ASN.1, TSN.1, EBML, etc.)
- Open « wishlist » : <https://dev.netzob.org>



Conclusion

- <http://www.netzob.org> , @Netzob
- Des questions à poser à Zoby ?



Comment contribuer :

- Partage de PCAPs (malware, scada...)
- Participation au développement
- Retour d'expérience



$$M(i,j) = \text{MAX} [M(i-1,j-1) + S(i,j) ; M(i,j-1) + W ; M(i-1,j) + W]$$

$S(i,j)$ = SCORING SCHEME | W = GAP PENALTY

	G	A	A	T	T	C	A	G	T	T	A
G	0	0	0	0	0	0	0	0	0	0	0
G	0										
A	0										
T	0										
C	0										
G	0										
A	0										

	G	A	A	T	T	C	A	G	T	T	A
G	0	0	0	0	0	0	0	0	0	0	0
G	0										
A	0										
T	0										
C	0										
G	0										
A	0										

	G	A	A	T	T	C	A	G	T	T	A
G	0	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1	1
A	0	1	1	1	1	1	1	1	1	1	1
T	0	1	1	1	1	1	1	1	1	1	1
C	0	1	1	1	1	1	1	1	1	1	1
G	0	1	1	1	1	1	1	1	1	1	1
A	0	1	1	1	1	1	1	1	1	1	1

	G	A	A	T	T	C	A	G	T	T	A
G	0	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1	1
A	0	1	2	2	2	2	2	2	2	2	3
T	0	1	2	2	3	3	3	3	3	3	3
C	0	1	2	2	3	3	4	4	4	4	4
G	0	1	2	2	3	3	4	4	5	5	5
A	0	1	2	3	3	3	4	5	5	5	6



Traceback step

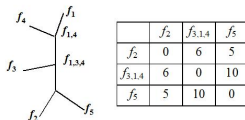
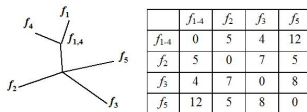
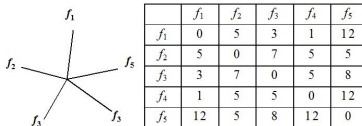
	G	A	A	T	T	C	A	G	T	T	A
G	0	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1	1
G	0	1	1	1	1	1	1	2	2	2	2
A	0	1	2	2	2	2	2	2	2	2	3
T	0	1	2	2	3	3	3	3	3	3	3
C	0	1	2	2	3	3	3	4	4	4	4
G	0	1	2	2	3	3	3	4	4	5	5
A	0	1	2	3	3	3	3	4	5	5	6

	G	A	A	T	T	C	A	G	T	T	A
G	0										
G		1									
A			1								
T				2	2						
C					3						
G						4	4				
A							5	5	5		
										6	

G A A T T C A G T T A
| | | | | | |
G G A _ T C _ G _ _ A



- Exécution itérative des opérations suivantes :
 - 1 Recherche dans la matrice le score le plus élevé.
 - 2 Regroupe les deux groupes associés.
 - 3 Mise à jour de la matrice (calcul de la moyenne des deux lignes).





Annexes

Comment fonctionne l'algorithme L*a

- Tant que l'automate hypothèse n'est pas conforme
 - Tant que la table n'est pas close et consistante
 - Soumission d'une requête d'appartenance
 - Construction d'un automate hypothèse
 - Recherche d'un contre exemple (requête d'équivalence)

