



# Modélisation et simulation du canal de communication d'un botnet pour l'évaluation des NIDS

Georges Bossert <sup>1</sup> <sup>2</sup>, Guillaume Hiet <sup>2</sup>, Thibaut Henin <sup>1</sup>

<sup>1</sup> AMOSSYS SAS - Rennes, France

<sup>2</sup> Equipe SSIR (EA 4039), Supélec

18 - 21 mai 2011



- 1 La détection des botnets
  - Les botnets
  - Problématique de l'évaluation
- 2 La modélisation pour la simulation
  - Le principe
  - La MMSTD
  - L'apprentissage du modèle
- 3 Implémentation et résultats
  - Implémentation
  - Résultats
- 4 Conclusion

- ▶ **BOTNETS :**
  - ▶ Ensemble de **systèmes interconnectés** interagissant pour accomplir des tâches distribuées
  - ▶ Contrôlés par une personne (ou un groupe) au travers d'un canal de communication appelé **C&C**
- ▶ **Les solutions pour s'en protéger :**
  - ▶ Analyse de l'impact du malware sur le système
  - ▶ Analyse de sa prolifération et de la topologie réseau
  - ▶ Analyse des symptômes réseau : détection du canal de communication
    - ▶ BotSniffer, BotHunter,
    - ▶ Snort, Bro, Suricata,
    - ▶ ...

- ▶ **BOTNETS :**
  - ▶ Ensemble de **systèmes interconnectés** interagissant pour accomplir des tâches distribuées
  - ▶ Contrôlés par une personne (ou un groupe) au travers d'un canal de communication appelé **C&C**
- ▶ **Les solutions pour s'en protéger :**
  - ▶ Analyse de l'impact du malware sur le système
  - ▶ Analyse de sa prolifération et de la topologie réseau
  - ▶ Analyse des symptômes réseau : détection du canal de communication
    - ▶ BotSniffer, BotHunter,
    - ▶ Snort, Bro, Suricata,
    - ▶ ...

- ▶ **BOTNETS :**
  - ▶ Ensemble de **systèmes interconnectés** interagissant pour accomplir des tâches distribuées
  - ▶ Contrôlés par une personne (ou un groupe) au travers d'un canal de communication appelé **C&C**
- ▶ **Les solutions pour s'en protéger :**
  - ▶ Analyse de l'impact du malware sur le système (**antivirus**)  
**[Out-of-Scope]**
  - ▶ Analyse de sa prolifération et de la topologie réseau
  - ▶ Analyse des symptômes réseau : détection du canal de communication
    - ▶ BotSniffer, BotHunter,
    - ▶ Snort, Bro, Suricata,
    - ▶ ...

- ▶ **BOTNETS :**
  - ▶ Ensemble de **systèmes interconnectés** interagissant pour accomplir des tâches distribuées
  - ▶ Contrôlés par une personne (ou un groupe) au travers d'un canal de communication appelé **C&C**
- ▶ **Les solutions pour s'en protéger :**
  - ▶ Analyse de l'impact du malware sur le système (**antivirus**)  
[Out-of-Scope]
  - ▶ Analyse de sa prolifération et de la topologie réseau (**réputations, black&white listes...**)[Out-of-Scope]
  - ▶ Analyse des symptômes réseau : détection du canal de communication
    - ▶ BotSniffer, BotHunter,
    - ▶ Snort, Bro, Suricata,
    - ▶ ...

- ▶ **BOTNETS :**
  - ▶ Ensemble de **systèmes interconnectés** interagissant pour accomplir des tâches distribuées
  - ▶ Contrôlés par une personne (ou un groupe) au travers d'un canal de communication appelé **C&C**
- ▶ **Les solutions pour s'en protéger :**
  - ▶ Analyse de l'impact du malware sur le système (**antivirus**) [Out-of-Scope]
  - ▶ Analyse de sa prolifération et de la topologie réseau (**réputations, black&white listes...**) [Out-of-Scope]
  - ▶ Analyse des symptômes réseau : détection du canal de communication **Cas d'utilisation typique d'un NIDS**
    - ▶ **BotSniffer, BotHunter,**
    - ▶ **Snort, Bro, Suricata,**
    - ▶ ...

## Comment évaluer l'efficacité des NIDS pour détecter ce type de menace ?

- ▶ Exécuter un malware sur un réseau ouvert mais contrôlé
  - ▶ Le réseau de commande n'est pas statique
  - ▶ Dépendance forte entre l'évaluation et le botmaster
  - ▶ Nombreux risques de fuites et d'attaques involontaires (coûts élevés x nombre d'évaluation)
- ▶ Déploiement du botnet en laboratoire
  - ▶ Très compliqué (coûts élevés x nombre d'évaluation)
  - ▶ Problème de réalisme

À considérer pour une méthodologie d'évaluation des **NIDS**

- ▶ Le **réalisme** assure l'**exactitude** de l'évaluation
- ▶ La **contrôlabilité** assure la **reproductibilité** de l'évaluation



## Comment évaluer l'efficacité des NIDS pour détecter ce type de menace ?

- ▶ Exécuter un malware sur un réseau ouvert mais contrôlé
  - ▶ Le réseau de commande n'est pas statique
  - ▶ Dépendance forte entre l'évaluation et le botmaster
  - ▶ Nombreux risques de fuites et d'attaques involontaires (coûts élevés x nombre d'évaluation)
- ▶ Déploiement du botnet en laboratoire
  - ▶ Très compliqué (coûts élevés x nombre d'évaluation)
  - ▶ Problème de réalisme

### À considérer pour une méthodologie d'évaluation des **NIDS**

- ▶ Le **réalisme** assure l'**exactitude** de l'évaluation
- ▶ La **contrôlabilité** assure la **reproductibilité** de l'évaluation

▶ **BESOIN** : Génération d'un trafic réseau

▶ **SOLUTION** :

- ▶ Rejeu de pcaps : réaliste mais incontrôlable
- ▶ Trafic synthétique : irréaliste mais contrôlable
- ▶ Méthode hybride : réaliste et contrôlable
  - ▶ Modélisation du trafic au travers de captures réseaux
  - ▶ Paramétrage des couches réseaux avec le modèle pour générer un trafic

▶ **BESOIN** : Génération d'un trafic réseau

▶ **SOLUTION** :

- ▶ Rejeu de pcaps : réaliste mais incontrôlable
- ▶ Trafic synthétique : irréaliste mais contrôlable
- ▶ Méthode hybride : réaliste et contrôlable
  - ▶ Modélisation du trafic au travers de captures réseaux
  - ▶ Paramétrage des couches réseaux avec le modèle pour générer un trafic

▶ **BESOIN** : Génération d'un trafic réseau

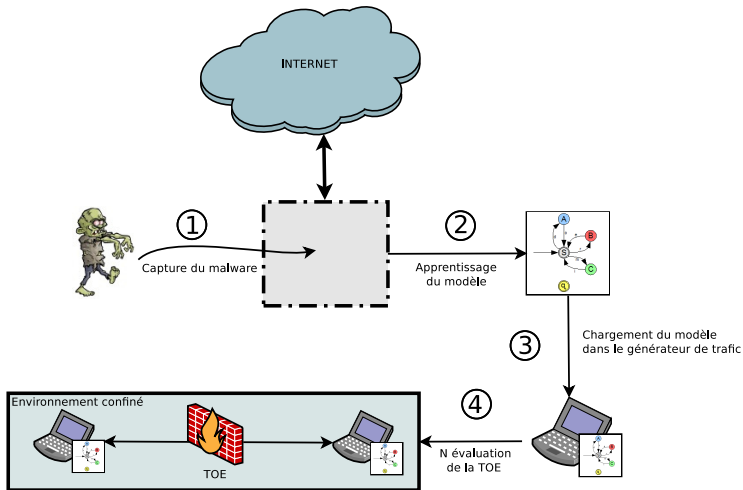
▶ **SOLUTION** :

- ▶ Rejeu de pcaps : réaliste mais incontrôlable
- ▶ Trafic synthétique : irréaliste mais contrôlable
- ▶ Méthode hybride : réaliste et contrôlable
  - ▶ Modélisation du trafic au travers de captures réseaux
  - ▶ Paramétrage des couches réseaux avec le modèle pour générer un trafic

▶ **BESOIN** : Génération d'un trafic réseau

▶ **SOLUTION** :

- ▶ Rejeu de pcaps : réaliste mais incontrôlable
- ▶ Trafic synthétique : irréaliste mais contrôlable
- ▶ Méthode hybride : réaliste et contrôlable
  - ▶ Modélisation du trafic au travers de captures réseaux
  - ▶ Paramétrage des couches réseaux avec le modèle pour générer un trafic



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

$q_0$  État initial

$S$  Ensemble des états

$X$  Alphabet des messages d'entrés

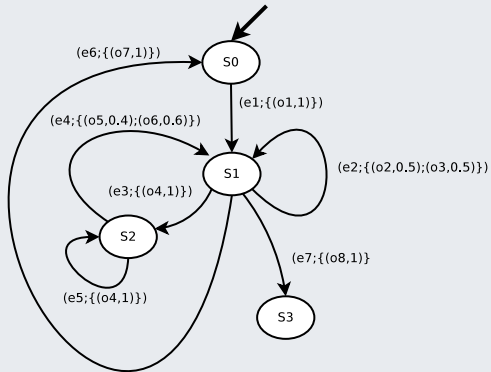
$Y$  Alphabet des messages en sorties

$T$   $|T| = |X| \times |Y|, T = \{A(y|x)\}$

Pour résumer :

- ▶ **Transitions déterministes** mais messages de sorties indéterministes
- ▶ Prise en compte du **temps de réaction**
- ▶ **Réduction** de l'alphabet d'entrée (\$EMAIL, \$DATE...)

### Exemple d'une MMSTD



#### 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"



## Apprentissage du modèle en 4 étapes

Etape 1 : Capture d'un C&C réel (tcpdump)

Etape 2 : Extraction de l'alphabet d'entrée, du temps de réaction et des séquences

Etape 3 : Inférence de la topologie du zombie confiné

Etape 4 : Généralisation des messages de sorties

## Équivalences assurées :

- ▶ Syntaxique (vocabulaire)
- ▶ Sémantique (grammaire)
- ▶ Temporelle (temps de réaction)

## Apprentissage du modèle en 4 étapes

Etape 1 : Capture d'un C&C réel (tcpdump)

Etape 2 : Extraction de l'alphabet d'entrée, du temps de réaction et des séquences

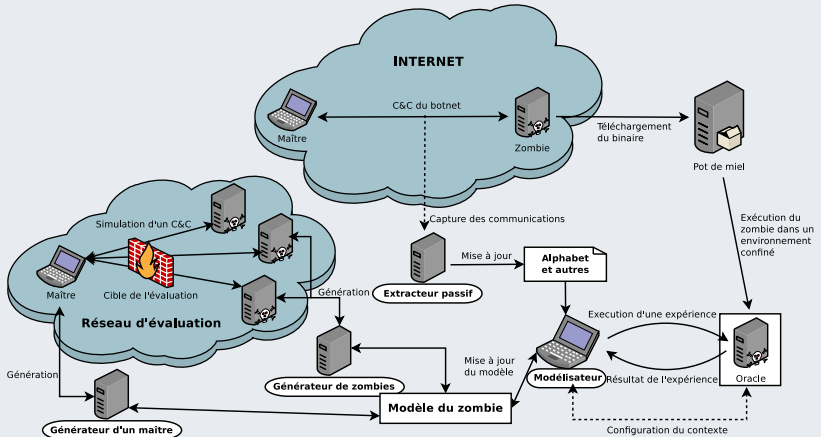
Etape 3 : Inférence de la topologie du zombie confiné

Etape 4 : Généralisation des messages de sorties

## Équivalences assurées :

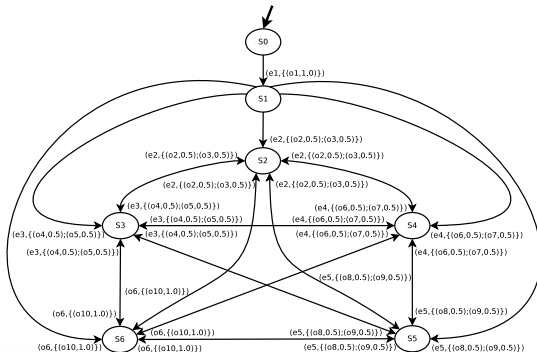
- ▶ Syntaxique (vocabulaire)
- ▶ Sémantique (grammaire)
- ▶ Temporelle (temps de réaction)

### Architecture



### Résultats

- ▶ Modèle du C&C du malware « pBot v2.0 by M@t »
- ▶ Validation de la détection du simulateur par un NIDS



#### Alphabets :

##### Entrées

e1="eh cik"  
 e2="info"  
 e3="pscan \$IP"  
 e4="download SURL"  
 e5="dns sURL"  
 e6="bot"

##### Sorties

o1="[Auth: LOGIN.....!!!!!!!!!!!!!!!!!!!!!!"  
 o2="[Info: Linux gbt-laptop 2.6.32-28-generic"  
 o3="[Info: Linux gbt 2.6.32-28-generic"  
 o4="[pscan: port open"  
 o5="[pscan: port close"  
 o6="[download: Nao foi possivel  
     fazer o download. Permissao negada."  
 o7="[download: Archivo baixado"  
 o8="[dns] www.google.fr => 74.125.230.80"  
 o9="[dns] www.google.fr => www.google.fr"  
 o10="[pbot] pphbot 2.0 recording by M3"

## Pros

- ▶ Caractérisation du vocabulaire et de la grammaire
- ▶ Automatisable
- ▶ Rapide (plusieurs heures) et réactif
- ▶ Simple à partager et à reproduire

## Cons

- ▶ Le chiffrement du C&C
- ▶ La complétude du modèle n'est pas assuré
- ▶ L'auto-protection (contre le sandboxing)
- ▶ Évolution du comportement (mise-à-jour)

## Travaux en cours & perspectives

- ▶ Automatiser la collecte de malware
- ▶ Créer un « repository » de modèles,
- ▶ Prendre en compte les actions du malware,
- ▶ Étendre le modèle avec d'autres caractéristiques ...



Questions ?

### Apprentissage de la grammaire

- ▶ Méthode : « l'élève et le professeur »
- ▶ Algorithme du  $L^* a$  (Angluin)
- ▶ Recherche des séquences de symboles valides
- ▶ Ré-initialisation entre chaque soumission (virtualisation)

