

Environnement graphique pour la programmation et le déploiement d'agents SCAPY

Encadrement : Rushed Kanawati
LIPN - CNRS UMR 7030
rushed.kanawati@lipn.univ-paris13.fr

1 Contexte

SCAPY est un logiciel libre de manipulation de paquets réseaux développé en langage `python`¹. Ce logiciel permet, entre autres tâches, d'intercepter le trafic sur un segment réseau et de générer des paquets et des trames dans un nombre important de protocoles (de tout niveaux). C'est un logiciel, libre disponible sur : <http://www.secdev.org/projects/scapy/>.

La possibilité de forger des paquets à la carte et de programmer des réactions à des paquets interceptés permet d'imaginer de scénarios pédagogiques innovants pour l'enseignement des réseaux informatiques. Cependant la nécessité de maîtriser un langage de programmation, en occurrence ici `python`, est une difficulté à surmonter. L'objectif de ce projet est de proposer un environnement de programmation graphique et intuitif qui faciliterait l'utilisation de SCAPY.

2 Travail à réaliser

Le but de ce projet est de développer un environnement graphique pour la programmation et le déploiement d'agents SCAPY. Il s'agit de développer des outils graphiques qui permettent de :

¹<http://www.python.org/>

- Forger à la carte, des messages (paquets et trames) de différents niveaux. On s'intéresse principalement à offrir la possibilité d'éditer des messages manipulés par les protocoles suivants : **ethernet** (802.3), **IPv4**, **ICMPv4**, **UDP**, **TCP**. La prise en compte des protocoles **IPv6** et **ICMPv6** est un plus.
- Offrir des modèles graphiques de génération de paquets réutilisables et paramétrables (i.e. supporter le concept de fonction).
- Spécifier des agents **SCAPY**. Il s'agit d'offrir la possibilité de spécifier le comportement d'un agent en termes de paquets à générer et à intercepter et avoir des actions planifiés d'une manière temporelle ou en réaction à des événements extérieurs (principalement l'interception de messages). Le langage graphique de script doit offrir les structures de contrôle de base : **instructions conditionnelles**, **boucle for**, **boucle while** en plus des opérateurs temporels (ex. temporisateurs, planification d'actions) et la programmation événementielle. On peut s'inspirer fortement du langage graphique **scratch** ² pour la conception des opérateurs graphiques.
- Offrir la possibilité de sauvegarder, charger et déployer les agents programmés.

3 Pourquoi choisir ce projet

- Pour maîtriser Python un langage fort utile pour le prototype rapide.
- Rafraîchir (!) vos connaissances en réseaux informatiques.
- Relever le défi de la conception d'un langage graphique et intuitif.
- Contribuer à améliorer l'approche pédagogique de l'enseignement des réseaux.

²<http://scratched.media.mit.edu/>