République **A**lgérienne **D**émocratique et **P**opulaire
Ministère de l'**E**nseignement **S**upérieur et de
la **R**echerche **S**cientifique
Université Ibn Khaldoun – Tiaret –

Faculté des Mathématiques et de l'informatique

Département de l'informatique

SPECIALITE: Informatique

OPTION : Informatique Répartie et Mobile (IRM)

MEMOIRE EN VUE DE L'OBTENTION DU DIPLÔME DE MAGISTER
(Ecole Doctorale)

Présenté par ***Mohamed Amine DAOUD***

Sujet du mémoire :

La Méthodologie de Modélisation et de la Simulation des cyber-attaques des réseaux par les variantes DEVS

Soutenu ***LE 09 Avril 2016***

Composition du jury

Mr. *Rachid CHALAL*	Professeur (*ESI*) - Alger -	Président
Mr. *Omar NOUALI*	Directeur de recherche *CERIST*-Alger-	Examinateur
Mr. *Amar BALLA*	Professeur (*ESI*) - Alger -	Examinateur
Mr. *Youcef DAHMANI*	Maître de conférences A (*UIK*) -TIARET-	Directeur du mémoire

ANNEE UNIVERSITAIRE 2015/2016

Résumé

A l'heure actuelle, *« tout est disponible partout à la fois »*, cela nous amène à s'interroger sur la question de la sécurité de l'information et de sa transmission pour éviter de compromettre un système d'information. Des attaques malveillantes pouvant porter atteinte à des propriétés essentielles telles que la confidentialité, l'intégrité ou la disponibilité des systèmes d'information. Dans un environnement de cyber-attaque, il y a de nombreuses entités qui sont impliquées, celles-ci ont des comportements et des ressources différents. Les cyber-attaques sont ici pour engendrer des pertes de données et des informations qui sont de plus en plus importantes pour les entreprises et utilisateurs finaux. Ces attaques sont élevées ces dernières années par une myriade des ressources infectées et comptent surtout sur les réseaux pour être contrôlés, se propagés ou endommagés. Face à ces risques, il y a besoin de manifester dans la réponse à ces nombreuses attaques par des stratégies de défense efficaces parmi eux, Nous trouvons la modélisation et la simulation. La modélisation et la simulation ne cessent de s'imposer comme des outils incontournables pour analyser le comportement des systèmes complexes. Afin de concevoir des modèles pour différentes attaques, un modèle généralisé et global ne peut représenter les différentes attaques et les entités inhérentes.

L'objectif du sujet est de modéliser et de simuler les cyber-attaques, par le formalisme à événements discrets DEVS et de ses différentes variantes connu par son expressivité et sa généralisation de modèles. Ce dernier est également connu par sa modularité et son hiérarchie. Pour mieux comprendre, nous le décrivons par ces deux contributions ;

La première contribution de ce travail est de proposer un nouvel objectif pour une taxonomie pragmatique pour les cyber-attaques. La construction de la taxonomie repose sur des propriétés dynamiques des cyber-attaques basée sur la chronologie des attaques. La classification est composée de quatre catégories nommées dans l'ordre : vecteur d'attaque, la cible, les types et les résultats d'attaques.

Une deuxième contribution est de proposer un modèle de base pour la structure des réseaux qui est de conception hiérarchique et modulaire. Ce modèle est construit par l'application de la structure de l'entité de systèmes (SES). Cette contribution ne sert pas à diviser la composante de la structure de l'entité de systèmes des cyber-attaques entre le matériel et logiciel mais d'avoir représenté tous les composants qui construisent les réseaux informatiques ainsi quelles sont leurs menaces parce que dans les derniers temps, les cyber-attaques sont bien diversifiées et complexes.

***Mots-clés** : Cyber-attaque, Modélisation, Simulation, DEVS.*

Dédicace

A mes parents pour tous leurs amours et leurs soutiens. Quoique je puisse dire, je ne peux exprimer mes sentiments d'amour et de respect à vos égards et ma gratitude.

A ma sœur, mes frères, ma grand-mère, votre aide, votre générosité, votre soutien ont été pour moi une source de courage et de confiance.

A la mémoire de ma Tante Nacira, mon oncle Lahcen et mon frère ABAID Kadda, que le destin ne nous a pas laissé le temps de jouir de ce bonheur ensemble et de vous exprimer tout mon respect.

Qu'il me soit permis aujourd'hui de vous assurer mon profond amour et ma grande reconnaissance.

A tous ceux qui comptent pour moi... Malgré la distance sans vous, aucune réussite n'aurait été possible. Aucun mot ne saurait retranscrire ici le bonheur que vous m'avez toujours apporté.

Je crois que vous êtes fiers de moi, autant que moi de vous.

Mohamed Amine DAOUD

Remerciement

C'est avec un grand plaisir que je réserve cette page en signe de gratitude et de profonde reconnaissance à tous ceux qui ont bien voulu apporter l'assistance nécessaire au bon déroulement de ce travail.

J'aimerais, tout d'abord, exprimer toute ma gratitude à mon encadreur Dr ***DAHMANI Youcef*** pour la qualité et la complémentarité de son encadrement et pour son soutien scientifique et sa disponibilité.

Je remercie Mon collègue ***BOUGUESSA Abdelkader*** de m'avoir aidé à bien mener ces conseils et son temps partagé avec moi et pour les multiples remarques très constructives et d'avoir trouvé le temps et la patience de m'aider dans mes travaux.

J'adresse mes sincères remerciements à tous les membres du jury qui m'ont fait l'honneur d'accepter de prendre part à ce jury et surtout de lire et d'expertiser mon travail. D'avoir accepté de participer au jury.

Je remercie les enseignants de l'Ecole nationale Supérieure de l'Informatique (ESI) et de l'université de Tiaret pour m'avoir transmis leurs savoir-faire et leurs connaissances.

Je souhaite remercier toutes les personnes ***Sonia***, ***Nassima***, ***Houda, Kadirou, Fadhallah, Yacine*** et ***Hocine*** avec lesquelles je travaille pour leur encouragement durant cette période particulièrement pour leurs contributions. Pour leurs aides précieuses, ainsi que les différents services qui m'ont permis de travailler dans d'excellentes conditions.

Merci aussi à tous mes collègues. Je leur exprime ma profonde sympathie. Toute mon amitié à mes amis qui se reconnaîtront ici, merci pour les moments d'amitié que nous avons partagés. Vous avez toujours été présents pendant les moments les plus difficiles de ma vie professionnelle et personnelle, j'ai eu beaucoup de chance de vous avoir comme amis.

Merci d'avoir cru à ce travail, merci de l'avoir défendu.

Table des matières

Introduction générale 1

CHAPITRE *I Etat de l'art*

1. Introduction 3
2. Cyber-attaques 3
   - *2.1.* Historique des cybers-attaques 3
   - 2.2. Définition cyber-attaque 5
   - *2.3.* Objectifs des cyber-attaques 6
   - *2.4.* Méthodologie d'une cyber-attaque 7
     - 2.4.1. Reconnaissance 8
     - 2.4.2. Numérisation 9
     - 2.4.3. Obtenir l'accès 10
     - 2.4.4. Maintenir l'accès 10
     - *2.4.5.* Couvrant les pistes ou Nettoyage des traces 11
   - *2.5.* Type des cyber-attaques 12
     - 2.5.1. Ingénierie sociale 12
     - *2.5.1.1. Le phishing ou hameçonnage* 13
     - 2.5.2. Infection informatique 14
     - *2.5.2.1. Programme simples* 14
     - *2.5.2.2. Programme auto reproducteurs* 16
3. Cyber-sécurité 18
   - 3.1. Définition de la cyber-sécurité 19
   - 3.2. Fondamentaux de la cyber-sécurité 19
     - 3.2.1. Disponibilité 19
     - 3.2.2. Intégrité 20
     - 3.2.3. Confidentialité 20
     - 3.2.4. Identification et authentification 20
     - 3.2.5. Non répudiation 21
   - 3.3. Protection des infrastructures 21
   - 3.4. Les outils de protection 23
     - 3.4.1. Cloisonnement des environnements 24
     - 3.4.2. Détection des intrusions 24

3.4.3. Protocole IP sécurisé 24
3.4.4. Chiffrement des données 24
3.4.5. Contrôle d'accès 25
4. Les modèles d'attaque 25
4.1. Graphe d'attaque 25
4.2. Les arbres d'attaque 26
4.3. Réseau de Pétri 27
4.4. Les Réseaux bayésien 28
4.5. DEVS 29
5. Synthèses des différentes approches : 30
6. Conclusion 32

CHAPITRE *II Formalisme DEVS*

1. Introduction 33
2. Définition du DEVS 33
3. Modélisation DEVS 34
3.1. Modèle Atomique 34
3.2. Modèle Couplé 36
4. Simulation DEVS 38
5. Variantes de DEVS 38
5.1. Parallèle DEVS : 38
5.1.1. Le modèle atomique 39
5.1.2. Le modèle couplé 40
6. Conclusion 40

CHAPITRE *III Classification et modèle proposés*

1. Introduction 41
2. Analyse des classifications existantes 41
2.1. Les taxonomies en une seule dimension 41
2.2. Les taxonomies de multi-dimension 42
3. Caractéristiques de la taxonomie 43
4. Proposition d'une nouvelle classification 43
4.1. Le vecteur d'attaque (technique) 45
4.2. Le type (Méthodes) 45
4.3. La cible de l'attaque 45

4.4. Les résultats (l'impact de l'information) 46
5. Modèle Général pour la sécurité des réseaux 47
6. Conclusion 50

CHAPITRE IV Implémentation

1. Introduction 51
2. Description de l'attaque 51
2.1. Le débordement de la pile 52
3. Le Processus de l'attaque 52
3.1. Reconnaissance 52
3.2. Numérisation : 52
3.3. Accès au système 53
3.4. Maintien de l'accès : 55
4. Modélisation 56
5. Simulation 58
5.1. Simulateur de l'attaque de débordement 58
5.2. Simulateur Suite-DEVS 60
6. Conclusion 66
Conclusion Générale 67
Bibliographie 68

Table des Figures

Figure 1:Méthodologie d'une cyber-attaque 8
Figure 2: Types de cybers-attaques 12
Figure 3:Description d'un modèle atomique DEVS 35
Figure 4: Description de l'évolution des éléments d'un modèle atomique 36
Figure 5: Description d'un modèle couplé DEVS 37
Figure 6: Arbre de classe du simulateur DEVS 38
Figure 7: les transitions d'états en Parallèle DEVS 39
Figure 8: Classification proposée pour les cyber-attaques 46
Figure 9: Le modèle général pour les réseaux informatiques 49
Figure 10: étape 01 réception du script 53
Figure 11: Etape 02 détermination de @ de retour par Metasploit 54
Figure 12: Etape 03 Association de @ ESP au EIP 54
Figure 13: Etape 04 Exploit 55
Figure 14: modèle couplé de l'attaque de débordement du tampon 56
Figure 15: L'interface du simulateur de l'attaque de débordement 59
Figure 16: Un succès de la simulation de l'attaque 59
Figure 17: Un échec de la simulation de l'attaque 60
Figure 18: Le Suivi (Traking) de l'environnement Suite-DEVS 61
Figure 19: le modèle de l'attaque 63
Figure 20 : Injection des entrées 63
Figure 21 : Les changements les phases 64
Figure 22: les résultats de la simulation 64
Figure 23: les changements du modèle PASSIVE_MEMORY 65
Figure 24: les changements du modèle RUNNING_SHELL 65
Figure 25: les changements du modèle EXPLOIT 65

Liste des tableaux

Tableau 1:Comparaison entre les critères d'évaluation de la sécurité. 31
Tableau 2: Comparaison entre les critères d'évaluation de la sécurité 32

Introduction générale

Avec le développement rapide et inattendu des technologies, la plupart des individus et des organisations dépendent de l'information en ligne que ce soit dans leurs vies sociales ou professionnelles. La base de cette liaison a émergé les réseaux et plus particulièrement l'Internet. Par le développement de cette dernière, l'information globale est devenue une tendance générale pour l'évolution des ressources humaines. Maintenant, les systèmes d'informations font face à des cybers-attaques sophistiquées qui combinent des vulnérabilités pour la pénétration des réseaux, prennent des formes diverses et ils ont un large spectre d'effets. Une cyber-attaque est une attaque contre un système informatique ou un réseau, constitué d'un ensemble d'actions informatiques pour influencer le contrôle des communications, mettre en danger le fonctionnement des systèmes informatiques et par conséquence sur le comportement des humains.

Par leur multiplication et la complexité des réseaux, nous sommes obligés de faire et de maintenir la sécurité pour réduire le danger des attaques et que les systèmes informatiques doivent être protégés pour assurer les objectifs de sécurité (CIA) Confidentialité, Intégrité et la Disponibilité. La sécurité des réseaux ce n'est pas seulement de déterminer les vulnérabilités mais comment elles sont combinées pour mettre en œuvre une attaque. Puisque les cybers-attaques sont partout, il est nécessaire de les comprendre, et de trouver des méthodes pour examiner la nature de ces attaques. Le meilleur moyen, c'est de classer les attaques et préciser les mécanismes d'attaques. Faire confiance à un système informatique est vital pour prendre les mesures nécessaires au niveau sécuritaire afin de les utiliser dans un environnement sain. Les mesures de sécurités ont été suggérées en fonction du critère de conformité, de détection des intrusions, de la politique de sécurité et de la modélisation des attaques.

La modélisation et la simulation ne cessent de s'imposer comme des outils incontournables pour analyser le comportement des systèmes complexes. Elles sont utilisées pour développer les techniques et les politiques de sécurité contre les attaques et d'évaluer les risques. La simulation à événement discret permet de discrétiser un problème donné et d'obtenir une représentation exploitable du système étudié. Son intérêt apparait lorsque le phénomène simulé utilise des échelles de temps différents. Parmi les formalismes à événement discret, on trouve le DEVS (*Discete EVent specification System*).

Notre travail, c'est d'avoir une méthodologie pour modéliser et simuler les cyber-attaques par le formalisme DEVS afin de permettre un enrichissement des techniques et des méthodes existantes. Cela signifie de classer les attaques et préciser les mécanismes d'attaques. Ce document est structuré en quatre chapitres.

Le premier chapitre permet de décrire un contexte sur notre travail « les cyber-attaques ». Nous avons commencé par un état de l'art sur les cyber-attaques et la cyber-sécurité. Ensuite, une étude détaillée sur les différents modèles utilisés dans le domaine des cyber-attaques.

Le deuxième chapitre présente notre formalisme de modélisation et la simulation à événement discret *DEVS*.

Le troisième chapitre présente un aspect d'une proposition sur la taxonomie des cyber-attaques et nous avons essayé de développer un modèle général pour les réseaux informatiques.

Le quatrième chapitre décrit une étude de cas d'une cyber-attaque, le choix de l'attaque est de débordement du tampon. Nous avons développé notre simulateur pour la compréhension de l'attaque et effectué une modélisation de l'attaque par l'application du formalisme de modélisation et une simulation par l'outil Suite-DEVS.

Finalement, une conclusion générale qui répond aux points essentiels de notre travail et fournit des perspectives de travaux restants à être effectué.

1. Introduction

Les cyber-attaques sont bien réelles et diverses, même si nous manquons des détails précis et que nous sommes souvent réduit à de simples appréciations générales sur leur importance, sur le nombre des cyber-attaques et des perturbations techniques, ainsi que sur la nature et l'ampleur des dommages effectifs ou des dommages possibles, la tendance des dernières années est absolument claire: des états, des entreprises et des particuliers sont agressés et subissent des dommages via la cyber-attaque qui sont augmentées en quantité et en qualité.

Cette situation est une conséquence de la mise en réseau croissante des infrastructures TIC (*Technologie de l'Information et de la Communication*), de notre dépendance toujours plus grande à leur égard et de l'opacité des processus d'appui. La complexité croissante des systèmes d'information les rend de plus en plus sensibles aux erreurs et défaillances augmentant ainsi les possibilités de les attaquer. Il faut donc s'attendre à ce que les cyber-attaques deviennent toujours plus professionnelles et dangereuses et à part des cas connus, d'autres attaques ne seront pas annoncées, voire pas découvertes. En effet, nombre de cas resteront non divulgués afin de préserver la réputation des entreprises attaquées.

Dans ce chapitre, nous s'intéressons en premier lieu à l'état de l'art sur les cyber-attaques et la cyber-sécurité. Deuxièmement, nous présentons une étude détaillée sur les différents modèles d'attaque.

2. Cyber-attaques

2.1. Historique des cybers-attaques

Les attaques informatiques, dont la première connue date de 1988, se propagent via internet, une clé USB ou directement par un ordinateur pour empêcher, gêner ou détourner le fonctionnement des systèmes d'information ; autrement dit, elles visent à entraver l'utilisation d'un appareil informatique ou à voler des informations. Leurs cibles vont des particuliers aux grandes industries et aux gouvernements du monde entier. Depuis quelques années, les cyber attaques de grandes ampleurs se sont multipliées et ont démontrés qu'aucune structure n'était à l'abri [*DEN, 2006*].

Kosovo :

En 1999, pendant la guerre du Kosovo, des hackers serbes ont attaqué les systèmes informatiques de l'OTAN pour protester contre les bombardements alliés.

Estonie :

En 2007, des attaques ont visé les sites du gouvernement, les medias, les banques estoniens et les operateurs téléphoniques. Il s'agissait d'une attaque DoS (Deni of Service) : les sites ne

répondaient plus car ils étaient saturés par une multitude de demande de connections simultanées. Le cas de l'Estonie a marqué l'esprit presque toutes les démarches administratives se font par internet. Cette dématérialisation a aggravé l'impact des cyber-attaques en paralysant quelques temps le pays.

Géorgie :

En 2008, des pirates russes et pro-russe ont cyber-attaqué la Géorgie lors des affrontements contre la Russie. Les sites de gouvernement, des medias et des infrastructures stratégiques (relais de communications, centrales électriques, approvisionnements en eau, banque nationale) ont été touchés par une attaques de déni de services.

France :

En 2006, une attaque a visé des centaines de sites français dont le ministère de l'éducation nationale. Les hackers turcs protestent contre le projet de négociation du génocide arménien. Les sites étaient « défigurés » : le drapeau turc et des messages apparaissaient.

En 2011, le ministère de l'économie et des finances a été ciblé d'une cyber-attaque. L'objectif des agresseurs était vraisemblablement de se procurer des informations économiques et financière sur la France.

Royaume-Uni :

En 2010, le site internet de la royal Navy, la marine britannique, a été victime d'une attaque revendiquée par un groupe de hackers roumains, appelé TinKode. La technique utilisée par les pirates est appelée attaque par injection SQL : elle exploite la faille d'une application pour obtenir l'accès à des informations sensibles.

Etats Unis :

En 2008, selon le secrétaire adjoint à la défense américaine J. William, une clé USB insérée sur un ordinateur de l'armée au Moyen-Orient aurait permis à des services de renseignements étrangers de pénétrer le commandement central de l'armée américaine (Attaque sur une structure de commandement).

En 2009, des cyber-attaques ont visé des sites internet américains. Parmi eux les sites de médias de la maison blanche et du département d'état. Ils s'agissaient d'une attaque DoS : les sites ne répondaient plus car ils étaient saturés par une multitude de demandes de connections simultanées.

Corée de sud :

En 2009, la Corée du sud a été victime de cyber-attaque. Plusieurs sites ont été visés et rendus inaccessibles, dont ceux de la présidence sud-coréenne et de la défense.

En 2010, une attaque informatique via le virus *Stuxnet* a touché des infrastructures, notamment en Iran, en Indonésie. Ce virus a infecté un logiciel SCADA (Supervisory Control And Data Acquisition-- Contrôle de supervision et acquisition de données--, système de

contrôle industriel) de Siemens, utilisé pour contrôler des infrastructures industrielles parfois vitales (eau, gaz, carburant, raffineries). Ce virus peut modifier le comportement du système SCADA et entrainer des conséquences potentiellement dramatiques. C'est une attaque contre des entreprises et des bases de commandement.

Le ver *Flame*, découvert en 2012, constitue quant à lui le système d'espionnage informatique le plus sophistiqué jamais découvert à ce jour. Contrôlé à distance, il est, entre autres, capable de copier tous types de fichier, de mémoriser les frappes sur le clavier, de déclencher le micro et l'émetteur Bluetooth, et peut s'autodétruire à tout moment.

Etats Unis

En février 2014, les établissements américains du groupe de loisirs « Las Vegas Sands » sont victimes d'une cyber-attaque majeure incluant le piratage du réseau informatique, un vol massif de données confidentielles puis la mise hors service d'une partie importante du système d'information et de télécommunications. Le piratage serait attribué à un groupe de hackers iraniens et ferait suite à la suggestion publique en octobre 2013 du milliardaire Sheldon Adelson actionnaire majoritaire de Las Vegas Sands, de « raser » Téhéran sous le feu nucléaire.

En novembre et décembre 2014, Sony Pictures Entertainment est victime d'une très importante fuite de l'ensemble de ses données, qui sont révélées par à-coup et revendiqué par le groupe « Guardian of Peace ».

2.2. Définition cyber-attaque

Définition 1 :

Une cyber-attaque est l'exploitation délibérée des systèmes informatiques, les entreprises dépendent de la technologie et des réseaux. Les cyber-attaques utilisent un code malveillant pour modifier le code informatique, de la logique ou de données, résultant en des conséquences perturbatrices qui peuvent compromettre les données et conduire à la cybercriminalité, telles que le vol de l'information et de l'identité [*DEN, 2006*].

Définition 2 :

Une cyber-attaque est une tentative visant à saper ou compromettre le fonctionnement d'un système informatique, ou de tenter de suivre les mouvements en ligne des personnes sans leur autorisation. Les attaques de ce type peuvent être indétectables à l'utilisateur final ou l'administrateur réseau, ou conduire à une telle désorganisation totale du réseau qu'aucun des utilisateurs ne peut effectuer même les plus rudimentaires de tâches, en raison de la sophistication croissante de ces types d'attaques réseau [*UIT, 2006*].

Il est important de comprendre qu'une cyber-attaque peut être relativement inoffensive et ne cause aucun type de dommages aux matériels ou aux systèmes. C'est le cas avec le téléchargement clandestin de logiciels espions sur un serveur ou un disque dur sans la connaissance ou le consentement du propriétaire de l'équipement. Avec ce type de cyber-attaque, l'objectif principal consiste généralement à recueillir de l'information qui va suivre

les mouvements généraux et les recherches effectuées par les utilisateurs autorisés à copier et transférer des documents clés ou des informations qui sont enregistrées sur le disque dur ou un serveur.

Bien que le but ultime est soit de capturer et de transmettre des informations qui aideront le destinataire d'atteindre une sorte de gain financier, les logiciels espions fonctionnent discrètement en arrière-plan et soit hautement improbable pour prévenir toutes les fonctions habituelles du système d'avoir lieu. Cependant, une cyber-attaque peut être malveillante dans son intention. Cela est vrai avec des virus qui sont conçus pour désactiver la fonctionnalité d'un réseau ou même un seul ordinateur connecté à Internet. Dans les situations de cette nature, le but n'est pas de recueillir des renseignements sans que personne s'en aperçoive, mais pour créer des problèmes pour toute personne qui utilise le réseau ou ordinateurs connectés à ce réseau attaqué. Le résultat final peut être une perte de temps et de revenus et, éventuellement, l'interruption de la fourniture de biens et services à des clients de l'entreprise touchée par la crise. Beaucoup d'entreprises prennent aujourd'hui des mesures pour assurer la sécurité du réseau et constamment améliorent pour prévenir ces types d'attaques informatiques malveillantes.

2.3. Objectifs des cyber-attaques

Derrière chaque cyber-attaque, il y a des objectifs et des motivations mais les principales cibles des cyber-attaques sont les données ou les informations des sites web gouvernementaux, des institutions financières, des forums de discussion en ligne, des actualités et des sites web militaires. Les buts et motivations des cybers-attaques impliquent certains processus :

- *Obstruction de l'information* : L'objectif principal de l'attaquant est de bloquer l'accès à des informations importantes de n'importe quel endroit et n'importe quel moyen quand les utilisateurs ont besoin des données ou des informations particulières.

- *Retard de processus décisionnel* : Les cyber-attaques ont un rôle majeur pour paralyser des domaines essentiels dans la vie tels que les services d'urgences et militaires qui entrainent des retards dans le processus de prise de décision telles que l'activation de soutien de la vie et aussi dans des services nucléaires.

- *Refus de fournir des services publics* : Ce but est de bloquer les utilisateurs légitimes d'accéder à des informations de toute organisation ou institution relatives à des services publics. Les cybers-attaques peuvent causer des perturbations dans des domaines tels que : les banques et les marchés boursiers. (spécifique par rapport le premier point)

- *Réduction de la confiance du public* : Suivant les cybers-attaques et le vol de l'information donc il y a une perte de confiance entre l'organisation et le public sur la fiabilité et la sécurité des données.

- *Réputation du pays sera dénigré* : Attaqué une réputation d'un pays est un motif principal d'une cyber-attaque. En raison de l'évolution technologique, chaque pays dispose des compétences qui améliorent son prestige entre les différents pays en développement.

2.4. Méthodologie d'une cyber-attaque

Les cybers-attaques visent à voler des informations de n'importe quel endroit d'une organisation ou d'un gouvernement. Pour voler ou pirater des données ou des informations, le pirate suit certaines caractéristiques pour atteindre son objectif. Parmi les caractéristiques, nous citons : harmonisée, énorme, exigeante du temps et des ressources, organisé et non spontanée. Pour bien comprendre, nous essayons d'éclairer ces caractéristiques :

- *Harmonisé* : Pour infecter un système, le pirate doit attendre que le processus soit accordé. Cette synchronisation des étapes de piratage amène le pirate à réaliser ce qu'il attend et recevra des résultats dans les meilleurs délais.

- *Organisé* : Pour effectuer une opération de cyber-attaque, le pirate utilise des méthodes bien formées et organisées pour infecter les systèmes ciblés très facilement. L'utilisation de méthodes organisées permet à obtenir des résultats plus efficaces.

- *Enorme* : Le lancement des cyber-attaques est généralement à grand échelle et infecte des millions d'ordinateurs à travers la planète, provocant des dégâts de données à grande échelle et des pertes financières.

- *Non spontanée* : Les attaques, qui se produisent délibérément, sont méticuleuses et avec une planification minutieuse afin de provoquer un massacre.

- *exigence du temps et des ressources*: Pour faire une cyber-attaque, il faut à l'avance une bonne planification donc beaucoup de temps et d'argent pour organiser une attaque.

Dans cette partie, nous présentons la méthodologie retenue par les pirates pour accéder aux systèmes informatiques et suivant l'étude de cette méthodologie, nous pourrons donner une bonne sécurité de systèmes et de les protéger contre ces attaques utilisées par des pirates, il faut connaitre bien la cartographie des vulnérabilités des systèmes. [*UIT, 2006*]
Pour que les pirates puissent accéder aux systèmes informatiques d'un organisme ciblé, il faut au début chercher les failles dans ces systèmes, c'est-à-dire les vulnérabilités nuisibles à la sécurité de système que ce soit dans les protocoles, les systèmes d'exploitation ou des applications. Les termes *Vulnérabilité*, de *brèche* ou en langage plus familier de *trou de sécurité* sont également utilisés pour désigner les failles de sécurité.

Pour pouvoir mettre en œuvre un exploit (ce terme signifiant d'exploiter une vulnérabilité), la première des choses, le pirate récupère le maximum d'informations sur l'architecture du réseau, les systèmes d'exploitation et les applications fonctionnant sur celui-ci par l'utilisation des outils. Suivant cette première étape de collecte des informations, le pirate donc établit une cartographie sur l'architecture de système, il est en mesure d'appliquer des exploits relatifs aux versions des applications qu'il a recensées. Un premier accès à la machine lui permettra de développer son action et récupérer d'autres informations et d'étendre ses privilèges sur la machine. Une fois, le pirate a obtenu un accès administrateur (Root est généralement utilisé), nous parlons alors de compromission de la machine car les fichiers systèmes sont susceptible d'avoir été modifiés, le pirate possède un niveau plus haut de droit sur la machine. Lors de la dernière étape de ce processus et s'il s'agit d'un pirate professionnel, il consiste à effacer ses traces, afin d'éviter tout soupçon lié à ce pirate de la part de l'administrateur du réseau compromis et de telle manière à pouvoir garder le plus longtemps le contrôle cette machine.

Maintenant, nous détaillons chacune de ces étapes de la méthodologie d'une cyber-attaque mentionnée dans la figure (1).

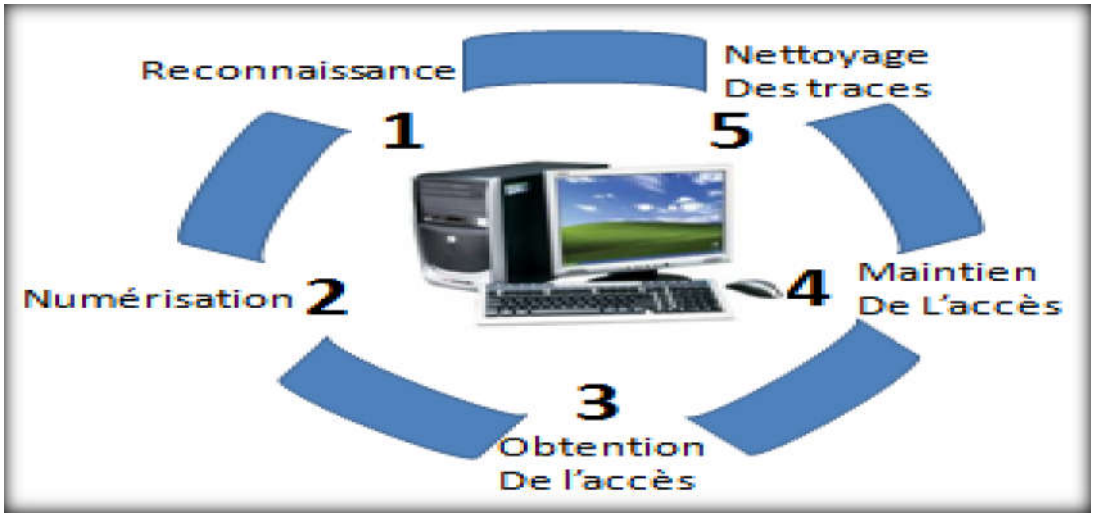

Figure 1:Méthodologie d'une cyber-attaque

2.4.1. Reconnaissance

Reconnaissance se réfère à la phase préparatoire où un attaquant rassemble autant d'informations que possible sur la cible avant de lancer l'attaque. Toujours dans cette phase, l'attaquant s'appuie sur la vie concurrentielle d'avoir plus d'informations sur la cible. C'est la phase qui permet à l'attaquant potentiel d'élaborer des stratégies de son attaque. Cela peut prendre un certain temps que l'attaquant attend pour dénicher des informations cruciales [*GER*]. Une partie de cette reconnaissance peut impliquer «l'ingénierie sociale». Un ingénieur social est une personne qui laisse parler des gens pour révéler des informations telles que les numéros de téléphone, mots de passe et autres informations sensibles.

Une autre technique de reconnaissance est « la fouille de poubelles ». Fouille de poubelles est le processus de recherche grâce à la poubelle de l'organisation des informations sensibles au rebut. Les attaquants peuvent utiliser l'Internet pour obtenir des informations telles que les

informations de contact des employés, les partenaires commerciaux, les technologies utilisées et d'autres connaissances critiques de l'entreprise mais « la fouille de poubelles » peut leur fournir des informations encore plus sensibles telles que : nom d'utilisateur, mot de passe, relevé de carte de crédit, relevé bancaire, numéro de sécurité sociale et numéro de téléphone. La sensibilisation de l'utilisateur des précautions qu'ils doivent prendre pour protéger leurs actifs d'information est un facteur essentiel dans ce contexte.

Types de reconnaissance :

Techniques de reconnaissance peuvent être classés en général en deux types : la reconnaissance active et la reconnaissance passive.

Reconnaissance passive : Quand un attaquant se rapproche de l'attaque en utilisant des techniques de reconnaissance passive, il n'interagit pas avec le système directement. Il utilise les informations disponibles publiquement, l'ingénierie sociale et la fouille de poubelles sont un moyen de recueillir des informations.

Reconnaissance active : Quand un attaquant emploie des techniques de reconnaissance active, il essaie d'interagir avec le système en utilisant des outils pour détecter les ports ouverts, des hôtes accessibles, l'emplacement des routeurs, la cartographie de réseau, les détails des systèmes d'exploitation et les applications.

2.4.2. Numérisation

La numérisation est un procédé que le pirate effectue avant d'attaquer le réseau. Dans l'analyse, l'attaquant utilise les informations recueillies lors des reconnaissances pour identifier les vulnérabilités spécifiques. L'analyse peut être considérée comme une extension logique (et recouvrement) de la reconnaissance active. Souvent les attaquants utilisent des outils automatisés tels que les scanners réseau/hôte et les composeurs de guerre pour localiser les systèmes et tenter de découvrir des vulnérabilités. Un attaquant peut recueillir des informations de réseau critique comme la cartographie des systèmes, des routeurs, des pare-feu et en utilisant des outils simples tels que *Traceroute*.

Scanners de ports peuvent être utilisés pour détecter les ports d'écoute à trouver des informations sur la nature des services en cours d'exécution sur la machine cible. Un attaquant suit une séquence particulière d'étapes afin de numériser n'importe quel réseau. Les méthodes d'analyse peuvent différer en fonction des objectifs d'attaque qui sont mis en place avant que les attaquants commencent effectivement ce processus.

Les outils les plus couramment utilisés sont les scanners de vulnérabilités qui peuvent rechercher plusieurs vulnérabilités connues sur un réseau cible et la possibilité de déceler des milliers de vulnérabilités. Ce qui donne à l'attaquant l'avantage de temps parce qu'il n'a qu'à trouver un moyen unique d'entrée tandis que le professionnel des systèmes doit garantir de nombreuses régions vulnérables en appliquant les correctifs. Les entreprises qui déploient des systèmes de détection d'intrusion ont encore des raisons de s'inquiéter parce que les attaquants peuvent utiliser des techniques d'évasion, tant au niveau de l'application et de réseau. La technique de défense primaire à cet égard est d'arrêter les services qui ne sont pas nécessaires.

Un filtrage approprié peut aussi être adopté comme un mécanisme de défense. Toutefois, les attaquants peuvent toujours utiliser des outils pour déterminer les règles mises en place pour le filtrage.

2.4.3. Obtenir l'accès

L'accès est la phase la plus importante d'une attaque en termes de dommages potentiels. Les pirates ne doivent pas toujours avoir un accès au système pour causer des dommages. Par exemple, les attaques par déni de service peuvent être soit échapper des ressources ou arrêter les services de s'exécuter sur le système cible, arrêter un service peut être effectué en tuant les processus, en utilisant une bombe logique ou même de reconfigurer le fonctionnement du système. Les ressources peuvent être épuisées localement en remplissant des liaisons de communication sortante. L'exploit peut se produire localement, en ligne, sur un réseau local ou l'Internet comme une tromperie ou de vol.

Les attaquants utilisent une technique appelée *Spoofing* pour exploiter un système en faisant semblant d'être des étrangers ou des systèmes différents [*GER*]. Ils peuvent utiliser cette technique pour envoyer un paquet malformé contenant un bogue dans le système cible afin d'exploiter la vulnérabilité. L'inondation de paquets peut être utilisée pour arrêter à distance la disponibilité des services essentiels.
Attaques *Smurf* ont essayé d'obtenir une réponse des utilisateurs disponibles sur un réseau puis utilisent leur adresse légitime pour inonder la victime [*GER*]. Les facteurs, qui influencent les chances d'un accès de gagner en attaquant un système cible, comprennent l'architecture, la configuration du système cible, le niveau de compétence de l'auteur et le niveau initial de l'accès obtenu. Le type le plus dommageable des attaques par déni de service peut être distribué par déni de service des attaques, où un attaquant utilise un logiciel de zombie distribué sur plusieurs machines sur Internet pour déclencher un déni orchestré à grande échelle de services.

2.4.4. Maintenir l'accès

Une fois, un attaquant a eu un accès sur les gains du système cible, l'attaquant peut choisir d' utiliser à la fois le système et ses ressources et utiliser davantage le système de rampe de lancement pour analyser et exploiter d'autres systèmes, ou à garder un profil bas et continuer à exploiter le système. Ces deux actions peuvent endommager l'organisation. Par exemple, l'attaquant peut mettre en place un renifleur pour capturer tout le trafic réseau, y compris les sessions *Telnet* et *FTP* (File Transfert Protocol) avec d'autres systèmes. Les attaquants, qui choisissent de passer inaperçu, suppriment la preuve de leur entrée et utilisent une porte dérobée ou un cheval de Troie pour accéder à répétition. Ils peuvent également installer des rootkits au niveau du noyau pour obtenir un accès super-utilisateur. La raison derrière cela est que les rootkits ont accès au niveau du système d'exploitation tandis que les gains des chevaux de Troie ont accès au niveau de l'application. Les rootkits et les chevaux de Troie dépendent des utilisateurs qui les ont installés. Dans les systèmes Windows, la plupart des

chevaux de Troie sont installés comme un service et exploités comme système local qui a un accès administratif.

Les attaquants peuvent utiliser des chevaux de Troie pour transférer des noms d'utilisateur, mots de passe et même des informations de carte de crédit stockées sur le système. Ils peuvent garder le contrôle sur leur système pendant une longue période par «durcissement» du système par rapport aux autres attaquants. Ils peuvent ensuite utiliser leur accès pour voler des données, de consommer de cycles CPU et le commerce des informations sensibles. Les organisations peuvent utiliser des systèmes de détection d'intrusion ou de déployer des pots de miel pour détecter les intrus. Ces derniers ne sont cependant pas recommandés sauf si l'organisation a un professionnel de la sécurité requise pour tirer parti de la notion de protection.

2.4.5. Couvrant les pistes ou Nettoyage des traces

Un attaquant aimerait détruire les preuves de sa présence et ses activités pour diverses raisons tel que le maintien de l'accès et de soustraire des mesures punitives. L'effacement des preuves d'un compromis est une obligation pour tout attaquant qui voudrait rester obscur. C'est l'une des meilleures méthodes pour échapper d'être retracé. Cela commence généralement avec l'effacement de connexions contaminées et les messages d'erreur possibles qui peuvent avoir été générés par le processus d'attaque. Par exemple, une attaque par débordement de tampon aura généralement laissé un message dans les logs du système. Ensuite, l'attention est tournée pour effectuer des changements afin que les futures connexions ne soient pas enregistrées. En manipulant les journaux d'événements, l'administrateur du système peut être convaincu que la sortie de son système est correcte, et qu'aucune intrusion ou compromis a effectivement eu lieu. Depuis, la première chose qu'un administrateur système fait pour surveiller l'activité inhabituelle est de vérifier les fichiers journaux du système, il est courant pour les intrus d'utiliser un utilitaire pour modifier les logs du système. Dans certains cas extrêmes, les Rootkits peuvent désactiver toute la journalisation et jetez tous les journaux existants. Cela se produit si les intrus ont l'intention d'utiliser le système pendant une période de temps plus longue que celle d'une base de lancement pour de futures intrusions. Ils permettent ensuite d'enlever uniquement les parties des feuilles qui peuvent révéler leur présence. Il est impératif pour les attaquants de restaurer le système à son état initial de l'accès et les portes dérobées rétablies pour leur utilisation. Tous les fichiers qui ont été modifiés doivent être changés à leurs attributs d'origine.

Ainsi, cette phase d'attaque peut se transformer en un nouveau cycle d'attaque en utilisant à nouveau des techniques de reconnaissance.

2.5. Type des cyber-attaques

Une attaque informatique peut être définie comme une action dirigée contre des systèmes informatiques dans le but de compromettre des équipements et l'exécution des processus ainsi que leur contrôle. Chaque type d'attaque vise des systèmes différents ou exploitants des différentes vulnérabilités multiples et implique l'utilisation d'armes adaptées dont certaines sont aujourd'hui entre les mains de groupes terroristes.

Dans [*BHAR, 2011*], il a traité trois types d'attaques contre les systèmes informatiques : physiques, syntaxiques et sémantiques.

- Une attaque *physique* implique des armes conventionnelles dirigées contre des centres informatiques ou des ensembles de câbles assurant les liaisons telles que les bombes ou les incendies.
- Une attaque *syntaxique* utilise un logiciel de type virus pour perturber ou endommager un système ou un réseau informatique qui s'appelle aussi « les infections informatiques ».
- Une attaque *sémantique* est une approche plus exacte. Son objectif est d'attaquer la confiance des utilisateurs qui s'appellent aussi « l'ingénierie sociale ».

Dans la représentation des types d'attaques, nous s'intéressons aux deux derniers types (syntaxiques et sémantiques). Nous décrivons ces types d'attaques en dessous et qui sont représentés dans la figure 2 :

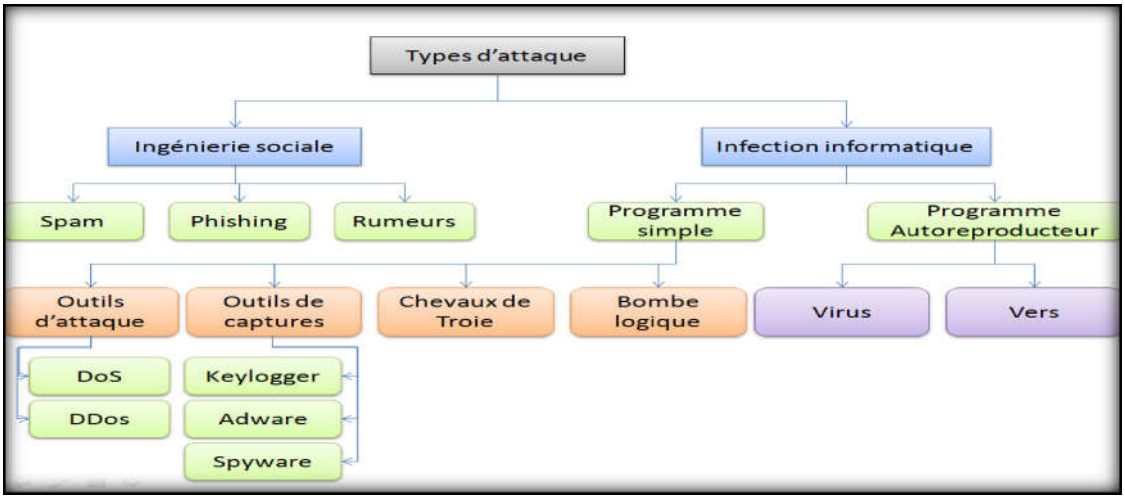

Figure 2: Types de cybers-attaques

2.5.1. Ingénierie sociale

Une démarche d'ingénierie sociale est l'utilisation des défaillances de comportement humain pour s'introduire dans un système d'information. Elle est à l'origine de très nombreuses fuites d'information des entreprises. Ce point critique doit être abordé afin que l'apprenant connaisse le principe et qu'il soit capable d'identifier ce type d'attaque que ce soit en contact physique direct ou via différents supports de communication (téléphonique, courrier, mail, hameçonnage, web, logiciel…).

L'ingénierie sociale n'est pas vraiment une attaque informatique. C'est plutôt une méthode pour obtenir des informations sur un système ou des mots de passe. C'est une approche psychologique en utilisant des acteurs humains disposants des informations pertinentes sur le système cible à attaquer. Pour atteindre le but, il y a 5 méthodes:

- Internet: Se faire passer pour un acteur de l'entreprise.
- Contact: Permet de cibler la personne et obtenir des informations dans son contexte (rendez-vous pour un motif quelconque) ou hors de son contexte (au restaurant d'entreprise)
- Téléphone: Préparer d'une identité, d'un rôle ou d'un but exprimé.
- Fax: Copier l'en-tête d'un fournisseur ou d'un client pour obtenir une information
- Lettre: Même approche pour le fax mais au lieu de copier l'en-tête, on utilise une adresse fictive.

2.5.1.1. Le phishing ou hameçonnage

Le hameçonnage est un acte d'un e-mail à un utilisateur prétendant faussement être une entreprise connue dans une tentative d'escroquerie à l'utilisateur en abandonnant l'information privée qui sera utilisé pour vol d'identité. L'e-mail dirige l'utilisateur à visiter un site Web où ils sont invités à mettre à jour des renseignements personnels. C'est une technique liée au monde de la finance. Il s'agit d'obtenir des données sensibles afin de commettre des impostures à l'identité et des escroqueries financières. L'imposture débute souvent par la réception d'un courrier non sollicité, l'escroc la réalise en 3 étapes :

- Il se fait passer pour ce qu'il n'est pas (une entreprise connue) pour solliciter les données convoitées auprès des internautes.
- Il présente des contenus fallacieux qui font illusion (faux liens, fausses pages web).
- Une fois les données convoitées, il se fait passer pour procurer des services ou des biens.

2.5.1.2. Rumeurs

Un *hoax* est une rumeur malveillante et non fondée qui est diffusée dans le but de leurrer ou de nuire. Dans la vie courante, les rumeurs ont toujours existées dans différents contextes sociaux, militaires, politiques ou économiques.

Les rumeurs qui perturbent le monde de l'informatique annoncent généralement l'apparition de nouvelles menaces imminentes et hautement destructrices qu'aucun outil de sécurité ne peut intercepter. De nombreuses rumeurs peuvent être regardées a posteriori, comme de simples farces. Cependant, propagées à grande échelle à travers des messageries à l'aide de liste de diffusion, elles peuvent perturber et bloquer temporairement le système de communication. Jusqu'à présent, la plupart de ces rumeurs sont identifiables à leur caractère excessif. Elles visent toutes à faire croire à la présence de faux virus sous une forme ou une autre. Il faut néanmoins s'attendre pour l'avenir à des morphologies de rumeurs beaucoup plus perverses et agressives. Plus difficiles à appréhender, certaines rumeurs d'aujourd'hui peuvent devenir demain des réalités.

2.5.1.3. Le courrier non sollicité (spam)

Il s'agit de l'envoi massif et parfois répété de courriers électroniques non sollicités à des personnes avec lesquelles l'expéditeur n'a jamais eu de contact, et dont il a capté l'adresse électronique de façon irrégulière. Nous savons tous qu'il est extrêmement facile et peu couteux d'atteindre certains d'individus à travers du courrier électronique. Cela n'a pas échappé aux publicitaires et à divers individus peu recommandable qui se cachent souvent derrière une adresse falsifiée pour inonder nos boites aux lettres. L'expéditeur ne connait pas les destinataires, il ne cible ni ses relations personnelles, ni ses relations professionnelles. Les adresses ont été collectées à grande échelle. Nous retrouvons principalement dans ces courriers :

- Des messages à caractères commercial.
- Des incitations à la visite de site web.

2.5.2. Infection informatique

Les infections informatiques sont des programmes ou des sous ensembles de programmes malveillants qui, à l'insu de l'utilisateur, sont destinées à perturber, modifier ou détruire tout ou une partie des éléments indispensables au fonctionnement normal de l'ordinateur. Nous différencions les programmes simples et les programmes autoreproducteurs.

2.5.2.1. Programme simples

Un programme simple contient une fonctionnalité malveillante (*Payload*) cachée qui se déclenche ou s'initialise lors de son exécution. Il n'y a pas de propagation. En un seul exemplaire, ce programme doit être introduit dans l'ordinateur ciblé. C'est souvent l'utilisateur lui-même qui, par manque de discernement, introduit le programme. Ce processus peut également être le travail d'un virus. L'action induite peut avoir un caractère destructif ou simplement perturbateur. Elle peut être immédiate ou retardée dans le temps. Dans de nombreux cas, le programme appelé s'installe à l'insu de l'utilisateur et modifie les paramètres du système pour qu'il puisse ensuite s'exécuter à chaque démarrage de la machine. Il agit alors de manière discrète et continue.

Nous retrouvons dans cette catégorie :

- Les bombes logiques
- Cheval de Troie
- Porte dérobée des outils de capture d'information
- Des outils d'attaque réseau etc…

a) Bombe logique :

C'est un programme contenant une fonction destructrice cachée et généralement associée à un déclenchement différé. Cette fonction a été rajoutée illicitement à un programme hôte qui conservera son apparence anodine et son fonctionnement correct jusqu'au moment choisi par

le programmeur malveillant. Elle peut être conçue pour frapper au hasard (aveugle) ou de manière ciblée.

- *Bombe logique ciblée* : un programmeur insère dans le programme de paie de l'entreprise qui l'emploie une fonction de destruction dont l'exécution est déclenchée si son nom disparait du fichier du personnel.
- *Bombe logique aveugle* : un programmeur insère dans un logiciel public distribué gratuitement sur internet une routine de destruction qui se déclenche chaque 1 avril.

b) Chevaux de Troie et portes dérobées (Backdoors) :

Ces programmes permettent d'obtenir un accès non autorisé sur les équipements qui les contiennent. Nous utilisons le terme de *cheval de Troie* lorsqu'il s'agit d'une fonction cachée et rajoutée au sein d'un programme légitime quelconque. Le terme *porte dérobée* s'applique à tout programme malveillant spécifiquement dédié à cet effet. Il s'agit en fait de l'un des éléments d'une d'application client/serveur permettant la prise de contrôle à distance d'un PC. Deux ordinateurs entrent en jeu. Le premier contient l'élément client, il pilotera le processus. Le second est la machine cible : il contient l'élément serveur – le cheval de Troie ou la porte dérobée. Il devra être actif sur la machine pour pouvoir initier la connexion avec le client. Le pirate interroge le réseau, à travers d'une adresse IP. Si celle-ci est joignable, la connexion s'effectue.

c) *Les outils de captures d'information* :

Les techniques de collecte d'information sont diverses. Il est possible de classifier les outils utilisés en fonction de l'information recherchée :

i. *Renifleur de clavier et de mot de passe* :

Un renifleur de clavier (*Keylogger*) est un programme permettant d'enregistrer les frappes au clavier. Son rôle ne se limite pas à l'enregistrement d'éventuel mot de passe. Il peut être sélectif ou enregistrer l'intégralité des informations qui transitent sur le périphérique de saisie. La plupart de ces dispositifs sont invisibles. Les frappes clavier sont généralement écrites dans un fichier temporaire chiffré et envoyé automatiquement, par courriel à l'espion. Beaucoup de virus actuels diffusent ces différents outils. Ils profitent du mode de propagation viral pour les installer plus aisément sur de nombreuses machines qui deviennent ainsi vulnérables à ce type d'attaque.

ii. *Espiogiciel* :

Au fil de la navigation sur le web, divers programmes sont installés sur l'ordinateur à l'insu de l'utilisateur. Ils sont plus communément connus sous leurs terminologies anglaises d'*Adware* et *Spyware*.

- Un *adware* (*advertising supported software*) est un logiciel qui permet d'afficher des bannières publicitaires. La plupart des annonceurs sont juridiques légitimes et leur

société commerciale reconnue. Les programmes ne diffusent pas d'information vers l'extérieur mais permettent la planification ciblée de message interne.

- Les spyware sont des *adware* qui installent sur le poste de l'utilisateur un logiciel espion et envoient régulièrement et sans accord préalable des informations statistiques sur les habitudes de celui-ci. Certains spyware ne se contentent pas de diffuser, mais ils modifient aussi les paramètres système à leur avantage pour imposer à l'utilisateur qui en est la victime, un certain mode de navigation sur le web. Ces logiciels peuvent aussi capturer vos habitudes en consultation hors ligne. Ils expédient les résultats de leur collecte à chaque ouverture du navigateur.

d) Outils d'attaque de réseau :

i. *Attaque en Déni de service* :

En terme de serveur, plus rarement de poste client, une attaque de type DoS est une activité consistant à empêcher quelqu'un d'utiliser un service. Pour ce faire, l'attaquant utilise un programme qui recherche à rendre le système ciblé indispensable en le faisant suspendre ou en le surchargeant.
En termes de réseaux, une attaque de type DoS consiste à submerger la victime d'un flot de trafic supérieur à sa capacité de traitement et le réseau devient indisponible.

ii. *Attaque en déni de service distribué* :

Il s'agit d'une attaque de type DoS qui utilise un grand nombre de machine simultanément. Ce type d'attaque se déroule généralement en deux temps. L'attaquant tente d'abord d'installer son outil dans le plus grand nombre de machine possibles. Celui-ci est programmé pour se déclencher soit à une commande (cas de BotNets), soit à un instant prédéfini. Il doit ainsi provoquer une surcharge bien importante que dans le cas d'une attaque unique.

2.5.2.2.Programmes auto reproducteurs

La finalité d'un programme auto reproducteur est identique à celle d'un programme simple. Il s'agit d'exploiter, de perturber ou de détruire. A sa première exécution, le programme cherche à se reproduire. Il sera généralement résident en mémoire et dans un premier temps discret. Si elle existe, la fonctionnalité malveillante s'effectuera dans un délai plus ou moins court et sur un critère quelconque prédéfini. Pour de nombreux virus la perturbation se limite à la reproduction et à tous les ennuis qu'elle engendre. Il n'y a pas à proprement parler de fonction malveillante.

Les vers et les virus forment à eux seuls la famille des programmes auto reproducteurs, nous les retrouvons au premier rang des infections informatiques. La distinction entre vers et virus est généralement acquise même si elle apparait parfois des plus fines :

- Un vers est un programme capable de fonctionner de manière indépendante. Il se propage de machine en machine à travers des connexions réseau. Un vers ne modifie aucun programme, il peut cependant transporter avec lui des portions de code qui pourront par la suite effectuer une telle activité virus par exemple. La terminologie

anglaise « Worm » est dérivée du mot « tape Worm » imaginé par John Brunner dans une de ses œuvres de science-fiction « sur l'onde de choc ».

- Un virus est un programme capable d'infecter d'autres programmes en les modifiant pour y inclure une copie de lui-même qui pourra avoir légèrement évolué. Le virus ne peut pas fonctionner d'une manière indépendante. L'exécution du programme hôte est nécessaire à son activation. Par analogie avec son cousin biologique, il se multiplie au sein de l'environnement qu'il cible et entraine corruption, perturbation ou destruction.

a) Les vers :

Il est possible de séparer les vers en deux grands groupes selon qu'ils utilisent les réseaux locaux ou qu'ils s'appuient sur Internet

i. Vers de réseaux locaux :

Il est facile de franchir le pas entre disques locaux et disques réseaux. La technique des vers de disquette s'est très vite étendue à l'ensemble des disques partagés. L'infection se déroule généralement de la manière suivante :

- Recherche de disque accessible ;
- Affectation de noms de lecteurs ;
- Copie du ver ;
- Exécution.

ii. *Vers de l'internet* :

Créés grâce à une parfaite connaissance de l'environnement réseau et l'utilisation de nouvelles failles de sécurité, ces nouveaux venus sont répertoriés parmi les plus dangereux. L'attaque touche ici les serveurs et non plus les stations de travail. Tout débute par l'exploitation d'une vulnérabilité. Celle-ci est généralement connue, mais comme les correctifs n'ont pas été appliqués sur de nombreuses machines, le nombre des serveurs vulnérables est suffisant pour une forte propagation.

b) Les virus :

Il existe quatre catégories principales de virus décrites dans [*CLUSIF, 2005*]. Elles ont chacune une cible bien précise :

- Les virus programme, dont le vecteur de contamination principal est constitué par les exécutables.
- Les virus système, dont le vecteur de contamination est le secteur de partition ou le secteur de démarrage (Boot sector).

- Les virus interprétés qui regroupent les virus macro sur les documents et les virus script utilisant un langage de programmation particulier qui se rapprochent de la programmation par lot.

De nombreux virus cumulent les cibles et renforcent ainsi leur capacité de contamination. Ils prennent alors les noms de virus multipartites ou multifonctions.

i. *Virus programme* :

Les virus programme cherchent à infecter les exécutables binaires compilés. Le principe de fonctionnement est le suivant :

- Le virus est présent dans un fichier exécutable.
- Lorsque celui-ci est exécuté, le virus choisit et contamine un ou plusieurs autres fichiers.
- Il agit généralement par ajout entrainant une augmentation de taille.
- Il se maintient et réside en mémoire, il infecte d'autres fichiers à l'exécution, ou simplement lors d'une manipulation.

ii. *Virus système* :

Les virus systèmes étaient -de loin- les plus répandus. Ils infectent les zones systèmes des disques durs ou des disquettes :

- Secteur de partitions (MBR, Master Boot Record) pour les disques durs.
- Secteur d'amorce (Boot, Dos boot Record) pour les disques durs et les disquettes.

Pour s'approprier l'un de ces 2 secteurs, le virus peut être introduit via un programme spécifique. Les acteurs ont cependant immédiatement compris qu'il était beaucoup plus simple de concevoir un virus directement sous la forme d'un secteur de démarrage de disquette. Le principe de fonctionnement adopté est le suivant :

- le virus est présent dans le secteur de démarrage d'une disquette.
- Il contamine le PC lorsque le BIOS exécute le code.
- Il déplace ou écrase le code original du BOOT ou du MBR du disque dur.
- Il remplace ce code par lui-même.
- Il sauvegarde éventuellement le code excédant (code complémentaire de virus) dans d'autres secteurs, libres ou occupés
- Dés lors et à chaque nouveau démarrage, il sera résident en mémoire et capable d'infecter d'autres disquettes sur un simple accès.

3. Cyber-sécurité

Tout comme les gouvernements et les entreprises doivent être conscients de la possibilité d'une cyber-attaque de se produire, les individus doivent aussi prendre des mesures pour protéger leurs ordinateurs et le matériel connexe de soutenir une attaque. Une mesure

préventive de base est d'assurer la qualité anti-virus et anti-logicielle espions, et le mettre à jour sur une base régulière.

Puisque la menace est évolutive, l'enjeu, aujourd'hui, n'est pas d'empêcher 100% les attaques mais d'en limiter leurs impacts, la nuisance pour le fonctionnement et les intérêts des organisations. Pour cela, elle doit évidemment avoir mis en place des systèmes de protection, mais également une capacité d'audit afin de pouvoir identifier rapidement les intrusions indésirables. Il faut savoir qu'un attaquant peut pénétrer un système et s'y installer pendant plusieurs mois voire des années sans que cela ne soit perceptible par l'entreprise. Ensuite, il s'agit d'être réactif afin de relativiser l'impact de l'attaque en ayant identifié sans tarder quel type d'informations a pu être récolté par les assaillants.
La gravité des cyber-attaques détermine le niveau d'intervention et les mesures d'atténuation nécessaires, c'est-à-dire la cyber-sécurité [*BHAR, 2011*].

3.1. Définition de la cyber-sécurité

La Cyber Sécurité est une branche de la technologie informatique connue sous le nom de la sécurité de l'information appliquée aux ordinateurs et aux réseaux. L'objectif de la sécurité informatique inclut la protection de l'information et des biens contre le vol, la corruption, ou d'une catastrophe naturelle, tout en permettant à l'information et à la propriété de rester accessibles et productives pour ses utilisateurs. Le terme de sécurité informatique du système signifie que le processus collectif, les mécanismes sur les informations sensibles et les services seront protégés par la publication, la falsification ou l'effondrement d'activités non autorisées ou des personnes indignes de confiance et d'événements imprévus. Les stratégies et méthodologies de la sécurité informatique diffèrent souvent de la plupart des autres technologies informatiques en raison de leur objectif un peu exclusif de prévenir le comportement indésirable de l'ordinateur au lieu d'activer le comportement d'ordinateur voulu. La finalité de la sécurité informatique est de garantir qu'aucun préjudice ne puisse mettre en péril la pérennité de l'organisation. Cela consiste à diminuer la probabilité de voir des menaces à se concrétiser, à en limiter les atteintes ou dysfonctionnements induits, et autoriser le retour à un fonctionnement normal à des coûts et des délais acceptables en cas de sinistre.

3.2. Fondamentaux de la cyber-sécurité

La cyber-sécurité vise la protection des renseignements, souvent le bien, le plus essentiel et le plus précieux qu'une entreprise puisse posséder. Les solutions de sécurité doivent contribuer à satisfaire les critères de base de la sécurité que sont la disponibilité, l'intégrité et la confidentialité (critères DIC). A ces trois premiers critères s'ajoutent ceux qui permettent de prouver l'identité des entités (notion d'authentification) et que des actions ou évènements ont bien eu lieu (notions de non répudiation, d'imputabilité voire de traçabilité) [*UIT, 2006*].

3.2.1. Disponibilité

La disponibilité est mesurée sur la période de temps pendant laquelle le service offert est opérationnel. Le volume potentiel de travail susceptible d'être pris en charge durant la période

de disponibilité d'un service, détermine la capacité d'une ressource (serveur ou réseau par exemple). La disponibilité d'une ressource est, en outre, indissociable de son accessibilité. La disponibilité des services, systèmes et données est obtenue, d'une part, par un dimensionnement approprié et une certaine redondance des éléments constitutifs des infrastructures et, d'autre part, par une gestion opérationnelle des ressources et des services.

3.2.2. Intégrité

Le respect de l'intégrité des données, traitements ou services permet d'assurer qu'ils ne sont pas modifiés, altérés ou détruits de façon intentionnelle ou accidentelle. Cela contribue à assurer leur exactitude, leur fiabilité et leur pérennité. Il convient de se prémunir contre l'altération des données en ayant la certitude qu'elles n'ont pas été modifiées lors de leur stockage ou de leur transfert. L'intégrité des données ne sera garantie que si elles sont protégées des écoutes actives qui peuvent modifier les données interceptées.

3.2.3. Confidentialité

La confidentialité est le maintien du secret des informations, des flux, des transactions, des services ou des actions réalisées dans le cyberespace. Il s'agit de la protection des ressources contre une divulgation non autorisée. La confidentialité peut être réalisée par la mise en œuvre de mécanismes de contrôle d'accès ou de chiffrement. Le chiffrement des données (ou cryptographie), contribue à assurer la confidentialité des informations lors de leur transmission ou de leur stockage en les transformant de façon à ce qu'elles deviennent inintelligibles aux personnes ne possédant pas les moyens de les déchiffrer.

3.2.4. Identification et authentification

L'authentification doit permettre de ne pas avoir de doute sur l'identité d'une ressource. Cela suppose que toutes les entités (ressources matérielles, logicielles ou personnes) soient correctement identifiées et que certaines caractéristiques puissent servir de preuve à leur identification. Tous les mécanismes de contrôle d'accès logique aux ressources informatiques nécessitent notamment de gérer l'identification et l'authentification des entités.

Les processus d'identification et d'authentification sont mis en œuvre pour contribuer à réaliser:

- la confidentialité et l'intégrité des données (seuls les ayant droits identifiés et authentifiés peuvent accéder aux ressources, contrôle d'accès et les modifier s'ils sont habilités à le faire);
- la non répudiation (les entités identifiées et authentifiées ont réalisées telle action), la preuve de l'origine d'un message, d'une transaction (une entité identifiée et authentifiée à effectuée une émission), la preuve de la destination (une entité identifiée et authentifiée est destinatrice d'un message).

3.2.5. Non répudiation

Dans certaines circonstances, il est nécessaire de prouver la réalisation de certains évènements (action, transaction). A la non-répudiation sont associées les notions de responsabilité d'imputabilité, de traçabilité et éventuellement d'auditabilité. L'établissement de la responsabilité nécessite l'existence de mécanismes d'authentification des individus et d'imputabilité de leurs actions. Le fait de pouvoir enregistrer des informations afin de pouvoir «tracer» la réalisation d'actions est important lorsqu'il s'agit de reconstituer un historique des évènements, notamment lors d'investigations en milieu informatique pour retrouver éventuellement l'adresse d'un système à partir de laquelle des données ont été envoyées par exemple. Les informations nécessaires à une analyse ultérieure (journalisation des informations) permettant l'audit d'un système doivent être sauvegardées. Cela constitue la capacité des systèmes à être audités (notion d'auditabilité).

3.3. Protection des infrastructures

La protection des infrastructures contre les cyber-attaques est d'un intérêt majeur pour les organisations. Bien que de nombreuses mesures aient été prises pour réduire les risques dans la cyber-attaque, il s'avère qu'elles ne suffisent pas à couvrir tous les cas. Du fait qu'une augmentation encore plus importante des perturbations et des attaques contre ces infrastructures soit envisageable.

L'organisation de la lutte des cyber-attaques passe par la mise en place d'outils de détection et de prévention depuis le poste de travail jusqu'à la passerelle Internet. Il est possible d'identifier quatre (04) niveaux concentriques de risque :

- Le poste de travail et les ressources propres à l'utilisateur.
- Les passerelles (réseau et messagerie).
- Les ressources partagées.
- Le monde extérieur, hors du périmètre de l'entreprise.

Les anti-virus comme seule parade aux virus informatiques actuels ne suffisent plus. Certains virus/vers utilisent de nouvelles méthodes de propagation et d'action :

- Ils ne résident qu'en mémoire vive et se propagent via le flux internet.
- Ils exploitent des failles liées au système d'exploitation et au réseau.
- Ils s'associent à des outils de piratage.
- Ils utilisent la technique du « spam » pour initialiser leur propagation.

Les particuliers et les entreprises doivent diversifier leurs dispositifs et améliorer ainsi leur niveau de sécurité. Ces autres outils sécuritaires deviennent au fil du temps indispensables. Cette défense en profondeur est aussi une opportunité pour :

- Une variété dans les ressources de sécurité avec des anti-virus de «moteur » différents entre ceux installés sur les postes de travail, les serveurs, les passerelles Internet quand la taille de l'entreprise devient conséquente ou que son activité en ligne est critique.

- Le déploiement d'une politique de correctifs qui intègre une phase de test, la surveillance des failles au niveau systèmes d'exploitations et applications mais aussi, qui prend en compte l'ensemble des équipements et ressources présents sur le réseau, routeurs, imprimantes et des solutions de sécurité.

a) Les ressources propres à l'utilisateur :

La protection du poste de travail est déterminante. Chaque poste de travail doit être muni de son anti-virus. Même si l'attaque pénètre la passerelle Internet dans un format non reconnu, même s'il n'est pas détecté sur le serveur, il doit être intercepté avant que l'utilisateur n'ait son poste infecté. Maintenir à jour la protection de la station est l'une des taches les plus ardues de l'administration système. Ceci est spécialement le cas sur les équipements nomades qui ne sont pas connectés en permanence au réseau. Les sociétés développent des anti-virus offrant leurs propres outils de déploiement.

b) *Les ressources partagées* :

L'application des mises à jour critiques, des correctifs « patch » et des mises à jour applicatives ou systèmes s'étendent aussi aux ressources partagées. Les administrateurs devront de préférence utiliser des outils dédiés qui en géreront le déploiement. Ces gestionnaires de correctif et de configuration aideront et superviseront tous les processus de mise en place. Il est très fortement déconseillé de naviguer sur Internet à partir d'un serveur. Ce procédé est très pratique pour récupérer un correctif, consulter une base de connaissances, mais il vaut mieux le faire d'une autre station, et obliger les intervenants extérieurs à faire de même.

c) *Passerelle Internet* :

La mise en place d'un anti-virus et d'architecture de pare-feu est ici de la plus grande importance. Ces équipements complèteront les dispositifs précédemment décrits en protégeant le réseau interne de l'entreprise lorsque celui-ci débouche vers l'extérieur.
Les anti-virus pour passerelles devront traiter le plus grand nombre de types de trafic (FTP, http, STMP..) et savoir analyser un large panel de formats de document. Ils décrient aujourd'hui de prendre en compte les flux de messagerie instantanée ou ceux des téléchargements poste à poste ou encore de la téléphonie sur IP.
Le contrôle de contenu est une solution de surveillance dédiée à la messagerie électronique et à la navigation Internet (http, FTP..). Outre le fait que certains de ces outils savent détecter les virus et autres codes malveillants, ils permettent une analyse lexicale par mots clés dans les mails ou dans les URL. L'installation d'un logiciel anti-spam permettra de bloquer ou de limiter la prolifération des messages non-sollicités ou les phénomènes de *mailbombing*. Fonctionnant comme des solutions anti virus ou anti-spam, les systèmes de détection d'intrusion (IDS) se réfèrent à une base de signatures d'attaques connues. Elles ne peuvent détecter que celles dont elles possèdent la signature.

Afin de donner à leur solution plus de réactivité lorsqu'une attaque surgit, certains chercheurs ont décidé de transformer leur offre en IPS (Intrusion Prévention System) et axent leur technologie vers la prévention proactive, capable de réagir en temps réel lorsqu'une anomalie est détectée ou qu'une intrusion est avérée. L'équipement fonctionne selon des règles de comportement et de signatures d'attaques : il surveille les attaques en dépassement de tampon (buffer over flow), les chargements en mémoire, les modifications critiques du système d'exploitation, l'utilisation excessive du CPU et la diminution soudaine de la bande passante.
Les équipements signalent des divergences par rapport au fonctionnement normal des éléments surveillés. Contrairement au pare-feu, qui traite des requêtes et les interdit, de tels systèmes les analysent de façon continue et ne réagissent qu'en cas d'anomalie.

d) *Le monde extérieur* :

Des scanneurs de vulnérabilité peuvent permettre de faire un audit en évaluant la résistance des machines au sein d'un réseau protégé. Un outil efficace doit savoir détecter les failles et préconiser des solutions. Au delà des dispositifs techniques de sécurité, il faut aussi utiliser toutes les dispositions contractuelles ou légales disponibles : des lois récentes précisent la responsabilité des acteurs, accroissent les obligations des fournisseurs en matière de traçabilité.

e) *La dimension humaine* :

L'homme se retrouve acteur et responsable à tous les niveaux. La formation et l'information doivent être au cœur du dispositif organisationnel. La sensibilisation n'est jamais définitivement acquise, elle doit faire l'objet de rappels périodiques et adaptés. Ces dispositifs doivent se concrétiser dans :

- Une politique de charte.
- Un règlement intérieur.

Il est possible de nommer des correspondants sécurité qui pourront servir de relais bidirectionnels dans leur environnement proche. En prise directe avec les utilisateurs, le centre d'assistance doit travailler en lien étroit avec l'équipe sécurité. Celle-ci doit savoir comment réagir face aux interrogations des utilisateurs et évaluer la pertinence d'une mise à jour forcée et anticipée des outils de protection. Elle doit aussi pouvoir évaluer un risque ponctuel imposant un changement temporaire du niveau de sécurité appliqué.

3.4. Les outils de protection

Assurer la sécurité des informations, des services, des systèmes et des réseaux consiste à réaliser la disponibilité, l'intégrité, la confidentialité des ressources ainsi que la non répudiation de certaines actions, ou l'authenticité d'évènements ou de ressources. La sécurité des informations n'a de sens que si elle s'applique sur des données et des processus dont on est sûr de l'exactitude (notion de qualité des données et des processus) afin qu'ils soient pérennes dans le temps (notion de pérennité des données et de continuité des services).

3.4.1. Cloisonnement des environnements

La séparation et le masquage d'un environnement privé vis-à-vis de l'internet public repose sur l'installation d'un ou plusieurs systèmes pare-feu (firewalls). Un firewall est un système qui permet de bloquer et de filtrer les flux qui lui parviennent, de les analyser et de les autoriser s'ils remplissent certaines conditions, de les rejeter dans le cas contraire. Le cloisonnement d'un réseau permet de constituer des environnements IP disjoints, en rendant physiquement indépendants les accès des réseaux que l'on désire séparer. Cela permet d'interconnecter deux réseaux de niveaux de sécurité différents. [*UIT, 2006*]

Le firewall constitue un des outils de réalisation de la politique de sécurité et n'est qu'un des composants matériel ou logiciel de sa mise en œuvre. En effet, un firewall ne suffit pas à bien protéger le réseau et les systèmes d'une organisation. Il doit être également accompagné d'outils, de mesures et de procédures répondants à des objectifs de sécurité préalablement déterminés par la politique de sécurité. L'efficacité d'un firewall dépend essentiellement de son positionnement par rapport aux systèmes qu'il doit protéger, de sa configuration et de sa gestion.

3.4.2. Détection des intrusions

Intrusions, incidents, anomalies doivent être détectés et identifiés au plus tôt de leur survenue et faire l'objet d'une gestion rigoureuse afin d'assurer le fonctionnement normal des systèmes et leur protection. Un incident est un évènement qui survient inopinément. Il est le plus souvent sans gravité en lui même mais il peut engendrer des conséquences graves. Une anomalie est une exception, elle peut induire un fonctionnement anormal du système d'information pouvant conduire à une violation de la politique de sécurité en vigueur. Elle peut être d'origine accidentelle (par exemple une erreur de configuration) ou volontaire (une attaque ciblée du système d'information). Une intrusion est une caractéristique d'une attaque et peut être considérée comme un incident ou une anomalie.
La détection d'intrusion est l'ensemble de pratiques et de mécanismes utilisés afin de détecter des erreurs qui pourraient conduire à des violations de la politique de sécurité et de diagnostiquer les intrusions et les attaques. Un système de détection d'intrusions (IDS – Intrusions Détection System) se compose de trois blocs fonctionnels essentiels: la collecte des informations, l'analyse des informations récupérées, la détection des intrusions et les réponses à donner à la suite d'une intrusion décelée. [*UIT, 2006*]

3.4.3. Protocole IP sécurisé

La prise en compte des besoins de sécurité ont conduit à la révision de la version 4 IP. C'est également afin de pouvoir, d'une part, de disposer d'une plage d'adresses plus importante et augmenter le nombre d'adresses internet disponibles et d'autre part, pour pouvoir faire une allocation dynamique de bande passante pour supporter des applications multimédias, que le protocole IP a fait l'objet d'une refonte connue sous le nom d'IPnG (Internet Protocol next Generation) ou IP version 6 (IPv6)

3.4.4. Chiffrement des données

La mise en œuvre de techniques de chiffrement permet de réaliser la confidentialité des données, de vérifier leur intégrité et d'authentifier des entités. Il existe deux grands types de

système de chiffrement de données: le chiffrement symétrique (à clé secrète) et le chiffrement asymétrique (à clé publique).

Divers algorithmes de chiffrement existent. Quelque soit leur mode opératoire (symétrique ou asymétrique), ils reposent sur l'usage de clés. Généralement leur degré de robustesse est lié à la capacité à gérer les clés de chiffrement de manière sécurisée, à la longueur de la clé (la longueur minimale de la clé est fonction du type d'algorithme), de la sécurité de la plateforme matérielle et logicielle dans laquelle les algorithmes de chiffrement sont implantés et s'exécutent.

3.4.5. Contrôle d'accès

Un mécanisme de contrôle d'accès logique aux ressources informatiques est basé sur l'identification des personnes, leur authentification et sur les permissions ou droits d'accès qui leurs sont accordés. Sur la base d'une identification authentifiée, le mécanisme de contrôle d'accès accorde, en fonction du profil de l'utilisateur, l'accès aux ressources sollicitées. Cela suppose que l'identification de l'usager (Gestion des identités), que les preuves de son identité (Gestion des preuves de l'identité) et que ses droits d'accès, soient correctement gérés (Gestion des autorisations)

Le profil de l'usager regroupe toutes les informations nécessaires aux décisions d'autorisation d'accès. Il doit être défini avec soin et résulte de la définition de la politique de gestion des accès. L'authentification permet de lier la notion d'identité à une personne. Les autorisations d'accès permettent de filtrer sélectivement les demandes d'accès aux ressources et aux services offerts via le réseau afin d'en accorder l'accès qu'aux seules entités habilitées. La vérification de l'identité dépend d'un scénario où le demandeur d'accès donne son identité et une preuve qu'il est censé être le seul à connaître ou à posséder (mot de passe, clé confidentielle, empreinte). Le service d'authentification procède à une comparaison de ces informations avec des données préalablement enregistrées dans un serveur d'authentification.

4. Les modèles d'attaque :

Les modèles d'attaque ont été largement utilisés dans la sécurité de l'information. Dans la plupart des cas, il se concentre sur la façon de documenter les attaques dans une forme structurée et réutilisable. Des travaux dans [*WANG, 2013*] ont montré que les modèles d'attaque sont le moyen le plus approprié pour soutenir la découverte et l'évitement de vulnérabilité de sécurité en comparant les modèles d'attaque avec des lignes directrices de programmation, langages de modèle, critères d'évaluation et les bases de données de vulnérabilité. Dans ce cas, nous s'intéressons aux modèles d'attaques les plus utilisées par les modélisateurs des cyber-attaques dans les cinq dernières années.

4.1. Graphe d'attaque

Les réseaux d'ordinateurs sont en croissance énorme que ce soit en nombre ou en complexité, l'évaluation de leurs vulnérabilités à l'attaque doit être très nécessaire. Une partie intégrante de la modélisation de la sécurité des réseaux est construit par les graphes d'attaque. Actuellement, les graphes d'attaque sont utilisés pour analyser les failles de la sécurité des

réseaux [*WANG, 2012*]. Le modèle des graphes d'attaque, c'est d'avoir un multiple de vulnérabilités qui peuvent être combinées pour une attaque. Les vulnérabilités représentent des états du système en utilisant une collection de condition liées à la sécurité. L'exploitation de la vulnérabilité est modélisée comme une transition entre les états du système. La composition successive d'exploits est considérée comme une agression. Plusieurs documents décrivent comment utilisés les graphes d'attaques [*LIU, 2013*], [*NOEL, 2008*].

Un graphe d'attaque fournit une description concise de tous les chemins possibles d'une attaque, qui commence à partir de chaque vulnérabilité exploitée et à la fin dans le réseau cible d'une attaque, ce qui est crucial pour un administrateur du système de comprendre la nature de la menace et de décider des contres mesures appropriées. Le graphe d'attaque réseau est un diagramme de transition d'état dans lequel chaque état désigne un attaquant, un défenseur et un système, et les transitions correspondent aux mesures prises par un attaquant qui conduisent à un changement dans l'ensemble de l'état du système.

Les graphes d'attaques sont utilisés pour déterminer si les états buts (objectifs) désignés peuvent être atteints par des attaquants qui tentent de pénétrer les réseaux informatiques à partir des états de départs initiaux. Pour cette utilisation, ils sont des graphes dans lequel le nœud de départ représente un attaquant à un emplacement spécifié dans le réseau. Les nœuds et les arcs représentent les actions que l'attaquant prend et le changement dans l'état du réseau causé par ces actions. Les actions impliquent généralement des exploits. Un graphe d'attaque plein montrera toutes les séquences possibles des actions des attaquants qui ont finalement conduit au niveau désiré de privilège sur la cible.

Les avantages des graphes d'attaque :

- Définir l'ensemble de tous les chemins vulnérables possibles dans lequel la sécurité peut être violée ;
- Fournir un contexte nécessaire pour répondre sur une réelle attaque ;
- Détermination des vulnérabilités dans un système par la définition du plus court chemin vers un nœud cible dans le graphe.

Inconvénient du graphe d'attaque

- Les graphes d'attaque ne sont pas garantis de trouver une solution optimale à chaque fois qu'ils exécutent ;
- Il est difficile de faire directement un graphe d'attaque complet ;
- La nécessité de maintenir les graphes d'attaque à jour par rapport au changement pour présenter une situation fidele à l'état courant des chemins d'attaques ;
- Les graphes d'attaque ne s'adaptent pas bien et ne peuvent pas être utilisés pour modéliser les grands réseaux.

4.2. Les arbres d'attaque

Les arbres d'attaque ont été introduits par Scheiner en 1998, pour la description d'un processus sur laquelle un attaquant peut exploiter un système. Les attaques d'arbres sont originaires du monde de l'analyse des défauts qui adopte une représentation arborescente des

dépendances entre les composants d'un système, des vulnérabilités et les actions effectuées par un attaquant. Les arbres d'attaques peuvent être utilisés pour capturer les étapes d'une attaque et leurs interdépendances. Les arbres d'attaque ont été les plus connus pour la représentation des cybers-attaques [*FOVINO, 2009*]. La racine de l'arbre représente le but ultime, tandis que les branches montrent toutes les séquences possibles de mesures d'actions vers l'objectif. Un attaquant pourrait imaginer en remontant l'arbre qu'il a atteint un nouveau sous-objectif à chaque nœud. L'approche de modélisation mise en œuvre dans AT (*Attack Tree*) est de visualiser une attaque comme une hiérarchie de sous-objectif menant à l'objectif ultime. Les nœuds peuvent être connectés par l'opérateur AND (ET) si elles sont toutes des conditions préalables au nœud parent et peuvent être connectés par l'operateur OR (OU) si ces nœuds représentent des méthodes alternatives pour atteindre au nœud parent. Le type dominant du modèle arbre d'attaque est l'approche de la logique booléenne arborescente. Les relations booléennes peuvent être représentées par des formes d'algèbre booléenne ou de type d'arête entre les nœuds. La représentation varie mais la sémantique reste relativement stable pour toutes les approches [*PAUL, 2014*].

Parmi les points forts des arbres d'attaque :

- Cet outil de modélisation s'avère être simple et facile à l'utiliser ;
- La représentation graphique des arbres d'attaque : ce qui est excellent pour la communication et la collaboration entre les parties intéressées ;
- La capacité de fournir rapidement des résultats significatifs à faible temps et d'efforts ;
- La capacité d'explorer et d'expérimenter un grand nombre de variable, seules ou en combinaisons pour explorer une attaque.

D'autres parts, les arbres d'attaque souffrent d'un ensemble de faiblesses, parmi lesquelles :

- La difficulté de maintenir les arbres d'attaque lorsque leurs nœuds augmentent ;
- La mauvaise gestion pour les objectifs de sécurité ;
- Ne modélisent pas les événements stochastiques non contrôlés et ne s'adaptent pas bien ce qui rend difficile l'exhaustivité et la cohérence ;
- Ne peuvent pas modéliser les multiples tentatives d'attaque et le rôle du temps;
- Ses capacités sont limitées par sa construction limite et sa nature statique.

4.3. Réseau de Pétri

Les réseaux de Pétri ont été crées en 1939 par Carl Adam Pétri dans le but de décrire les processus chimiques. Les réseaux de Pétri ont été introduits comme une technique de modélisation pour les systèmes concurrents et donnent une approche intuitive riche et graphique pour la modélisation, simulation et l'exécution. La flexibilité des réseaux de Pétri rend son utilisation abstraite pour le processus d'une attaque. Les réseaux Pétri sont des modèles discrets de comportements asynchrones concurrents, ce qui leur permet de modéliser les cyber-attaques. Les réseaux Pétri ont été utilisés pour la modélisation dans de nombreux domaines de recherche tels que la modélisation des attaques [*WANG, 2013*], la modélisation du conflit des réseaux [*ZAKR, 2011*] et dans la modélisation du comportement d'une attaque dans les systèmes vulnérables [*ELBO, 2012*]. Diverses caractéristiques contribuent à un tel

succès comprenant : la nature graphique, la simplicité du modèle et la modularité dans la conception.

C'est un langage de modélisation mathématique, il se compose des lieux (endroits), des transitions et des arcs qui les relient. Les arcs d'entrées relient les lieux avec les transitions, les arcs de sorties commencent à une transition et terminent à un lieu. Les lieux peuvent avoir des jetons. Les réseaux de Pétri sont assez bons pour la description et l'étude des systèmes qui sont caractérisés comme étant concurrents, asynchrones, distribués, parallèles, non déterministes et stochastiques [*ELBO, 2012*].

Le formalisme Pétri décrit quatre aspects : les états, les événements, les conditions et les relations entre eux. Lorsque la condition a été satisfaite, l'événement lié aura lieu, l'occurrence de l'événement va changer les états dans le système et provoque d'autres conditions à satisfaire.

Les avantages des réseaux de Pétri :

- Les réseaux de Pétri sont à la fois expressifs et bien concis ;
- Ils sont évolutifs et prennent en charge la concurrence, l'exhaustivité et la cohérence.
- Son utilisation est bien connue pour leurs capacités de graphe et d'analyse ;
- Ils s'adaptent au cyber-attaque.

Les inconvénients des réseaux de Pétri

- Les réseaux de Pétri ne sont pas très réalistes au cyber-attaque ;

4.4. Les Réseaux bayésiens

Le réseau bayésien permet d'effectuer diverses analyses qui offrent des perspectives utiles et intégrales pour la sécurité [*SHIN, 2014*]. Les réseaux bayésiens sont utilisés pour les évaluations des risques de la sécurité. Dans le guide de la sécurité des réseaux, la conformité est intrinsèquement qualitative et quantitative mais il est difficile de représenter la relation qualité quantitative. Les réseaux bayésiens sont utilisés souvent pour surmonter cette difficulté en convertissant la valeur qualitative à la valeur quantitative [*SHIN, 2014*].

Les réseaux bayésiens sont un type de réseau causal qui modélise les relations causales en société avec les probabilités. Les réseaux bayésiens peuvent servir à un graphe d'un ensemble ordonné de comportements observables de périphérique d'un système. Chaque nœud représente un périphérique ou sous-système qui est associé à un ensemble d'événement observable. Les nœuds du graphe sont disposés verticalement pour représenter une relation de cause à effet. Les nœuds au même niveau dans les réseaux sont indépendants les uns des autres. Les arêtes du réseau représentent des dépendances conditionnelles entre un ensemble de comportement de périphériques requis pour créer un scenario plus grand. Les arêtes, sont pondérées par un facteur ou une combinaison de facteur qui indique la force de la relation de cause à effet [*PAN, 2012*].

Les réseaux bayésiens sont représentés par un graphe orienté acyclique. Chaque réseau est constitué d'un ensemble de sommets et un ensemble d'arêtes. Les nœuds du graphe

représentent des événements observables tandis que les arêtes représentent des dépendances conditionnelles entre les nœuds parents et enfants. Chaque réseau bayésien possède un ensemble des nœuds discrets $X = \{X_1, X_2, , , X_n\}$. Les arêtes dirigées dans le réseau indiquent l'incidence éventuelle que le comportement d'un ou plusieurs nœuds sources aura sur le comportement du nœud enfant. Ceci est représenté par la distribution de probabilité conditionnelle $P(X_i|V_i)$ où V_i est un vecteur contenant les nœuds parents X_i. Chaque nœud X_i peut prendre un certain nombre de valeurs $x_{ij}\epsilon\{x_{i1}, x_{i2}, , , x_{im}\}$ où m est le nombre possible des valeurs.

Le réseau bayésien est un graphe acyclique dirigé par des arcs qui représentent les dépendances entre les nœuds et les variables à l'aide de théorème de bayes :

$$P(C|x) = \frac{P(C)\,P(x|C)}{P(x)} \qquad (1)$$

- *P (x)* est la distribution de probabilité de la variable *x* à l'ensemble population ;
- *P(C)* est la probabilité antérieure que certain échantillon appartienne à une classe ;
- *P (x | C)* est la probabilité conditionnelle à l'obtention de la valeur de variable *x* ;
- *P(C | x)* est la probabilité postérieure, que la valeur de la variable *x* appartienne à une classe dans une situation donnée.

Parmi les avantages des réseaux bayésiens, nous trouvons :

- La représentation graphique explicite et interprétable de la connaissance incertaine telle que leur sémantique qui est basée sur le concept de l'indépendance conditionnelle parce que ce sont des modèles probabilistes ;
- La théorie de la décision est naturellement applicable pour traiter les problèmes liés au coût ;

Parmi les inconvénients des réseaux bayésiens, nous citons :

- Ils n'ont pas la capacité de tracer des trajectoires individuelles pour chaque scenario du système [*PAN, 2012*] ;
- Ne spécifient pas comment sont calculées les valeurs des probabilités conditionnelles d'une attaque sur chaque nœud ;
- Les RB n'abordent pas le problème de la gestion des risques optimaux [*POOL, 2012*].

4.5. DEVS

Dans 1976, Ziegler a introduit le DEVS (*Discrete Event system Specification*) comme un formalisme abstrait pour la modélisation à événement discret et est un formalisme universel [*ZIEGLER, 1976*]. DEVS permet au modélisateur de s'abstraire totalement de l'implémentation des simulateurs mettant en œuvre la modélisation du système.

Un travail dans [KIM, 2012], utilise la spécification du système à événement discret (DEVS) pour simuler la cyber-sécurité, parce que la simulation progresse en se basant sur les interactions qui ont eu lieu pendant l'événement des attaques.

Les avantages du formalisme DEVS :

- DEVS est un formalisme abstrait indépendant de l'implémentation et par conséquent des environnements de simulation ;

- Il offre une vision modulaire et hiérarchique des systèmes dynamiques ;
- La fonction de transition interne, formalise et permet le comportement autonome du modèle. Les modèles peuvent ainsi évoluer et émettre des événements lorsque aucun événement externe n'est programmé pendant une certaine durée ;
- DEVS est fermé sous composition. Cela signifie que toute composition obtenue par couplage de composants spécifiés par le formalisme est elle-même spécifiée par le formalisme ;
- L'interprétation, dans le monde réel, des concepts d'abstraction et de modélisation est explicite ;
- Il garantit la cohérence de la représentation construite, et par conséquent il offre un modèle implémenté pour simuler des modèles élaborés.

5. Synthèses des différentes approches :

Après l'étude des différentes approches de la modélisation et de la simulation pour les cyber-attaques cités ci-dessus, nous avons pu faire une synthèse par la comparaison entre les approches suivant des critères pour l'évaluation de la sécurité décrits dans le premier tableau, Le choix des critères de comparaison a été vraiment difficile parce qu'ils existent une variété où il n'existe pas des travaux qui touchent tous ces critères, qui seront cités :

- *Analyse de coût* : c'est de calculer les mesures de renforcement les moins coûteux ou de trouver les mesures minimales pour la sécurité d'une attaque ;
- *Evolutivité* : c'est la quantité de travail pour créer le modèle qui sera proportionnel au nombre d'étapes d'attaque modélisé ;
- *Evaluation les risques* : ce critère peut être utilisé pour aider à localiser les points faibles dans la conception d'un système. Ainsi, estimer par évaluation l'impact des vulnérabilités et des attaques ainsi que l'efficacité des mécanismes de protection mis en œuvre ;
- *Optimisation* : c'est de fournir des connaissances relatives aux attributs qui font une attaque possible ;
- *Planification dynamique* : dans chaque attaque, il y a une probabilité d'apparition pour changer pendant la durée de vie d'un système en raison des nouvelles conditions.

	Graphe d'attaque [*NOEL, 2008*]	***Arbre d'attaque*** [*POOL, 2012*]	***Réseau de Pétri*** [*ZAKR, 2011*]	***Réseau Bayésien*** [*POOL, 2012*]
Evolutivité	x	X	✓	ND
Evaluation des risques	✓	✓	✓	✓
Analyse du cout	x	✓	ND	✓
Planification dynamique	x	X	ND	✓
Optimisation	✓	✓	✓	✓

Tableau 1:Comparaison entre les critères d'évaluation de la sécurité.

Notes : *X* : n'existe pas, : Existe ND : pas défini

Après l'étude des approches de modélisation et de la simulation des cyber-attaques récentes, nous avons pu faire une seconde comparaison entre les approches suivant des critères communs entre ces approches décrites en ligne du tableau 2 :

- *Action* : c'est l'information dynamique sur l'attaque et cette action se poursuit pour effectuer une technique d'attaque. C'est de déterminer un niveau désiré de privilège sur la cible.
- *Dynamique* : c'est qu'un système décrit une fonction de transition qui s'applique sur un état pour générer un nouvel état.
- *Modularité* : Chaque composant est indépendant et peut être considéré comme une entité à part entière ou comme un composant d'un système le plus grand ;
- *Construction* : c'est la méthode suivie pour composer une attaque par le choix d'une telle approche ;
- *Indépendance* : c'est que l'implémentation et par conséquent des environnements de simulation sont indépendants ;
- *Hiérarchique* : Un modèle est décrit selon des différents niveaux d'abstraction.

	Graphe d'attaque	***Arbre d'attaque***	***Réseau de Pétri***	***Réseau Bayésien***	***DEVS***
Actions	Oui	Oui	Oui	Oui	Oui
Dynamique	-	Faible	Forte	Forte	Forte
Modularité	Non	Non	Non	Non	Oui
Construction	Facile	Facile	Facile	Peu difficile	Facile
Indépendance	Non	Non	Non	Non	Oui
Hiérarchique	Forte	Forte	Forte	Forte	Forte

Tableau 2: Comparaison entre les critères d'évaluation de la sécurité

6. Conclusion

Nous nous sommes focalisés dans ce chapitre de l'état de l'art sur deux importantes notions : la cyber-attaque et la cyber-sécurité. Dans la première partie qui est les cyber-attaques, nous avons touché la méthodologie que doit suivre tout pirate pour attaquer une cible et aussi nous avons axé sur les différents types d'attaque. Dans la cyber-sécurité, nous avons focalisé notre travail sur les fondamentaux de la sécurité et aussi aux outils qui sont utilisés pour garantir la sécurité contre ces attaques.

Aussi, pour mieux comprendre et maintenir la sécurité, nous avons décrit les modèles utilisés pour la modélisation et la simulation des cybers-attaques, parmi ces modèles, notre choix est axé sur le formalisme DEVS qui sera présenté dans le prochain chapitre.

1. Introduction

La modélisation et la simulation ne cessent de s'imposer comme des outils incontournables pour analyser le comportement des systèmes complexes. Plusieurs méthodes ont été proposées pour améliorer le processus d'analyse de comportement de ces systèmes. Les propositions tentent d'atteindre des modèles plus réalistes, assez simples et fortement flexibles. Un système complexe est caractérisé par le regroupement de composants plus ou moins hétérogènes, soit à travers les modèles, ou soit au sein même d'un modèle. Afin de garantir un déploiement rapide et efficace ainsi qu'une homogénéité au sein des échanges entre ces différents composants, il faut qu'ils présentent tous la même interface au système et surtout la même logique de fonctionnement, et qu'ils disposent d'un langage commun afin d'unifier la description de tous les comportements possibles. Le formalisme DEVS permet, dans une certaine mesure, de répondre à cette préoccupation. Dans ce chapitre, nous s'intéressons au formalisme DEVS.

2. Définition du DEVS

Depuis les années 1970, des travaux formels ont été menés pour développer les fondements théoriques de la modélisation et de la simulation des systèmes dynamiques à évènements discrets. Le formalisme *DEVS* (*Discrete EVent system Spécification)* a été introduit par le professeur *B.P. Zeigler* [*ZEIGLER, 1976*] comme un formalisme abstrait pour la modélisation à évènements discrets. La modélisation à événement discret, ça veut dire qu'elle permet de discrétiser un problème donné et d'obtenir une représentation exploitable du système étudié [*ZEIGLER, 1988*].

Le formalisme DEVS est une approche de modélisation basée sur la théorie générale des systèmes, c'est un formalisme universel. DEVS permet au modélisateur de s'abstraire totalement de l'implémentation des simulateurs mettant en œuvre la modélisation du système. Plus précisément, c'est un formalisme modulaire et hiérarchique pour la modélisation, centré sur la notion d'état. Un système est représenté, pour sa forme structurelle, par deux types de modèles atomiques et couplés.

Le formalisme DEVS a été défini par P. Anglani [ANGLANI*, 2000*] comme "*une méthodologie universelle et générale qui fournit des outils pour modéliser et simuler des systèmes dont le comportement est basé sur des évènements*".

La modélisation consiste à interconnecter les modèles atomiques et les modèles couplés du système étudié afin de former un nouveau modèle décrivant le comportement du système étudié, c'est l'aspect fonctionnel. Les modèles atomiques sont les composants de base du formalisme, ils décrivent le comportement du système. Leur fonctionnement est proche de celui des *machines d'états*. Pour décrire un système plus complexe nous interconnectons plusieurs modèles atomiques pour former un modèle couplé. Ce nouveau modèle peut être utilisé comme modèle de base dans une description de plus haut niveau, c'est l'aspect hiérarchique du formalisme.

Au niveau de la structure du système, cette approche peut sembler statique ; le formalisme DEVS dans sa forme basique ne tient pas compte de l'évolution potentielle de la structure du

système, seul les états peuvent évoluer. Toutefois le formalisme a été étendu pour permettre ces changements de structure, et il en est, ou peut en être de même pour d'autres aspects. Le formalisme DEVS peut être vu comme un environnement de multi-modélisation regroupant de manière cohérente d'autres formalismes de modélisation basés eux aussi sur la théorie générale des systèmes et centrés sur les états. Sa capacité d'ouverture, au sens informatique, en fait un formalisme adapté à un grand nombre de domaines d'application [*BARROS, 1994*] [*UHMA, 2001*] [*NTAIMO, 2002*] [*TROC, 2003*].

Au niveau de la simulation, il permet l'analyse de systèmes complexes à évènements discrets décrits par des fonctions de transitions d'états, et des systèmes continus décrits par des équations différentielles [*ZEIGLER, 2000*]. A ce niveau, le principal avantage de l'approche tient au fait que, pour un modèle décrit suivant les spécifications du formalisme, les algorithmes de simulations sont générés automatiquement. Cela permet de s'abstraire totalement de l'implémentation des simulateurs lors de la phase de modélisation, ce qui conduit à une séparation explicite entre la modélisation et la simulation.

3. Modélisation DEVS

Le formalisme DEVS repose sur la définition de deux types de modèles : les *modèles atomiques* et les *modèles couplés*. Les modèles atomiques permettent de représenter le comportement de base du système. Les modèles couplés, quant à eux, sont définis par un ensemble de sous modèles atomiques et /ou couplés. Ils permettent de représenter la structure interne du système grâce à la définition de couplages entre modèles.

3.1. Modèle Atomique

Le modèle atomique peut être vu comme une machine d'états basée sur le temps. Il permet de décrire l'aspect fonctionnel ou comportemental du système. Le modèle atomique fournit une description autonome du comportement du système, défini par des états et des fonctions d'entrées / sorties et de transitions internes du modèle. L'évolution du modèle se fait par changement d'état en fonction de stimuli externes (via une entrée) ou internes (via une fonction de transition). Ces changements d'états ont pour but de déterminer la réponse comportementale du système à ces stimuli.

Le modèle atomique, figure 1, est caractérisé par la spécification suivante :

$$MA =< X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta > \qquad (2)$$

Avec

- $X = \{(P_{IN}, v) \mid P_{IN} \in Ports\ d'entrée, v \in X_{Pin}\}$: la liste des entrées du modèle, chaque entrée étant caractérisée par un couple (numéro du port / valeur) ;
- $Y = \{(P_{out}, v) \mid P_{out} \in Ports\ de\ sortie, v \in Y_{P_{out}}\}$: la liste des sorties du modèle, chaque sortie étant caractérisée par un couple (numéro du port / valeur) ;
- S : l'ensemble des états ou des variables d'états du système ;
- δ_{ext} : $Q \times X \rightarrow S$: la fonction de transition externe, où :
  - $Q = \{(Si, e) \mid Si \in S, 0 \leq e \leq ta(Si)\}$: l'ensemble des états $S_{\{1,2,\ldots,n\}}$;
  - e : est le temps écoulé depuis la dernière transition ; La fonction de transition externe spécifie comment le modèle atomique change d'état (passage de l'état

S_1 à l'état S_2 quand une entrée survient (évènement externe) avant que *ta* (S_1) ne soit écoulé.

- δ_{int} : $S \rightarrow S$: la fonction de transition interne. Elle permet de passer d'un état S_1 à l'instant t_1, à un état S_2 lorsqu'aucun évènement externe n'arrive durant le temps de vie de l'état *ta* (S_1);
- $\lambda : S \rightarrow Y$: la fonction de sortie ;
- *ta* : $S \rightarrow R+$: la fonction d'avancement du temps, ou le temps de vie de l'état *S* ;

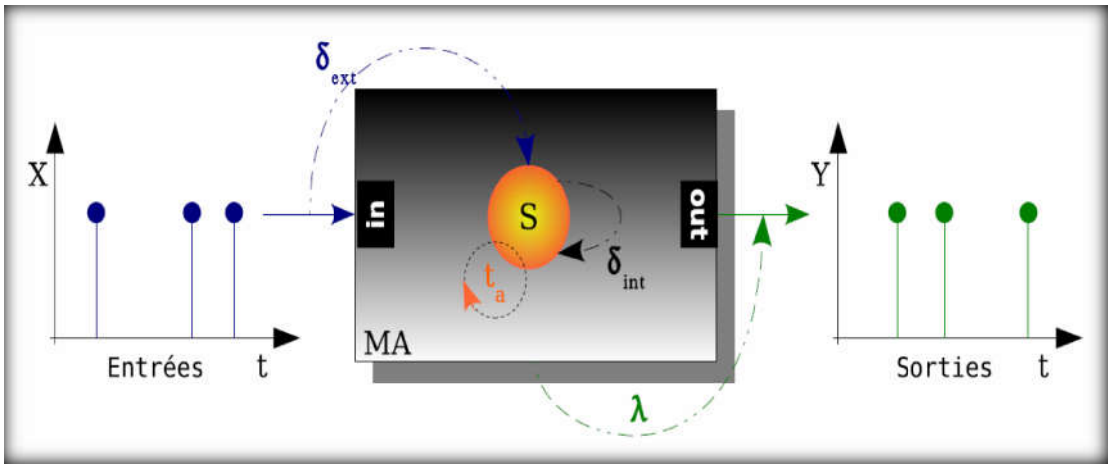

Figure 3:Description d'un modèle atomique DEVS

Les modèles atomiques réagissent à deux types d'évènements (stimuli) externes ou internes. Un évènement externe provient d'un autre modèle, il déclenche la fonction de transition externe (δ_{ext}) et met à jour le temps de vie de l'état (*ta*(*Si*)). Un évènement interne entraîne un changement d'état du modèle. Il déclenche les fonctions de transition interne (δ_{int}) et de sortie (λ). Le modèle calcule ensuite avec la fonction d'avancement du temps (*ta*) la date du prochain évènement interne. Ces enchaînements d'actions et la description du comportement du modèle sont présentés dans la figure (6). Elle présente l'évolution des états d'un modèle.

- *Xi* = {1 ,2} représente les entrées ;
- *Si* = {1, 2, 3} les états du modèle ;
- *e* l'écoulement du temps, il est remis à zéro à chaque changement d'état ;
- *ta* représente la durée de vie d'un état, elle est mise à jour après chaque changement d'état, si elle est égale à zéro la fonction de transition interne est déclenchée ;
- *Y* représente les sorties du modèle.

A chaque instant le modèle est dans un état (*Si* = {1, 2, 3}). Si un évènement externe *Xi* = {1, 2} $\in X$ est détecté avant que $e = ta(Si)$, le système change d'état grâce à la fonction de transition externe ($\delta ext(Si, e, Xi)$).

Dans la figure 4, nous passons de l'état S_1 à l'état S_2 lorsque l'entrée X_1 est détectée, puis de l'état S_1 à l'état S_3 lorsque X_2 est détectée à son tour. Si aucun évènement externe (*Xi*= {1,2}) n'est détecté, le modèle reste dans le même état pendant un temps donné par la fonction

ta(*Si*). Lorsque le temps de la vie de l'état est écoulé, c'est-à-dire lorsque $e = ta(Si)$ le système active sa fonction de sortie ($\lambda(Si)$). De plus, l'état du système est aussi mis à jour grâce à l'exécution de la fonction de transition interne ($\delta int(Si)$). Dans les deux cas, le système est dans un nouvel état (figure 4: $S1\ avec\ \delta int\ et\ S2\ et\ S3\ avec\ \delta ext$), avec un nouveau temps de vie et ainsi de suite.

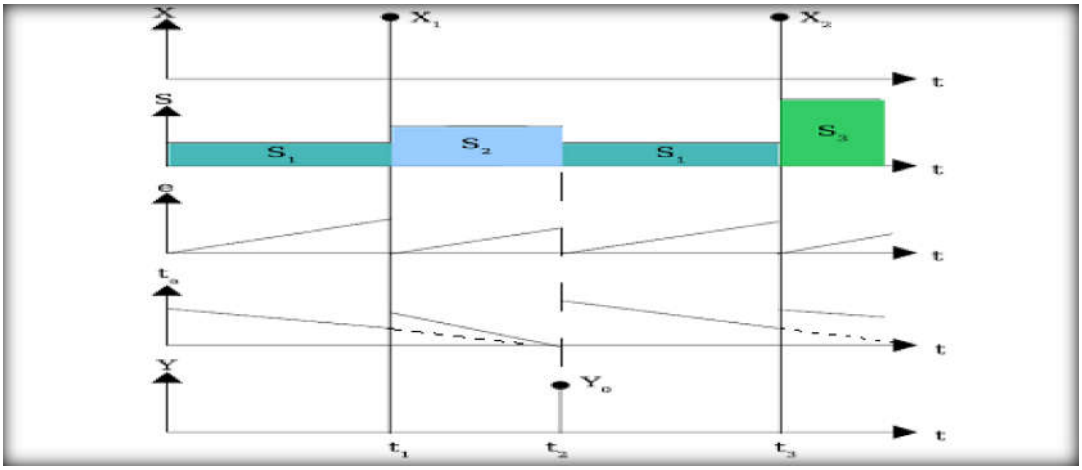

Figure 4: Description de l'évolution des éléments d'un modèle atomique

En fonction de son temps de vie, l'état d'un modèle peut être défini comme *transitoire* ou *passif*. Si *ta*(*S*) = 0 alors la durée de vie de l'état est tellement courte qu'aucun évènement externe ne peut intervenir avant l'arrivée du prochain changement d'état, le système est dans un état transitoire. Si *ta*(*S*) = ∞ le système restera dans le même état tant qu'aucun évènement externe n'est détecté, il est dans un état passif.
A partir de modèles atomiques nous pouvons représenter un grand nombre de systèmes en les interconnectant au sein d'un modèle couplé de plus haut niveau.

3.2. Modèle Couplé

Un modèle couplé DEVS est modulaire et présente une structure hiérarchique, ce qui permet la création de modèles complexes à partir de modèles atomiques et / ou couplés. Il est décrit par la formule :

$$MC =< X, Y, D, \{Mi\}, \{Ii\}, \{Zi, j\}, Select > \qquad (3)$$

Avec

- $X = \{(P_{IN}, v) \mid P_{IN} \in Ports\ d'entrée,\ v \in XP_{IN}\}$: la liste des entrées du modèle, chaque entrée étant caractérisée par un couple (numéro du port / valeur) ;
- $Y = \{(P_{out}, v) \mid p_{OUT} \in Ports\ de\ sortie,\ v \in YP_{out}\}$: la liste des sorties du modèle, chaque sortie étant caractérisée par un couple (numéro du port / valeur) ;
- D : la liste des modèles composant le modèle couplé *MC* ;
- $M_i =< Xi, Yi, Si, \delta ext_i\ \delta int_i, \lambda_i, ta_i >$: un modèle atomique ;
- Pour chaque modèle $i \in D\{MC\}$, *Ii* est l'ensemble des modèles qui influence i ;

- $Z_{i,j}$ est la fonction de transition des sorties du modèle *i* vers le modèle *j*, telle que :
  - $Z_{MC,j}$: *XMC* → *Xj* est la fonction de couplage des entrées externes (*EIC*) ;
  - $Z_{i,MC}$: *Yi* → *XMC* est la fonction de couplage des sorties externes (*EOC*) ;
  - $Z_{i,j}$: *Yi* → *Xj* est la fonction de couplage interne (*IC*) ;
- *Select* : la liste des priorités entre modèles.

- Une relation de couplage interne (*IC :Internal Coupling*) pour le couplage des ports des sous-modèles qui composent le modèle couplé ;
- Une relation de couplage des entrées externes (*EIC : External Input Coupling*) pour le couplage des ports d'entrée du modèle couplé avec les ports d'entrées de ses sous-modèles;
- Une relation de couplage des sorties externes (*EOC : External Output Coupling*) pour le couplage des ports de sortie du modèle couplé avec les ports de sortie de ses sous-modèles.

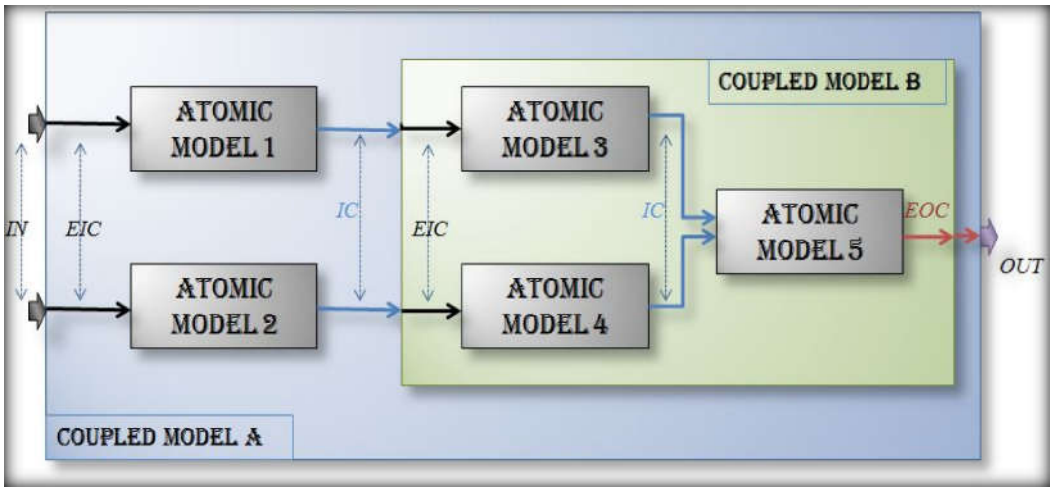

Figure 5: Description d'un modèle couplé DEVS

La Figure 5 décrit un exemple d'un modèle couplé (équation 2) est une composition de modèles atomiques et / ou de modèles couplés. Il présente un exemple de hiérarchie entre modèles, il est composé de deux modèles couplés (A et B) et cinq modèles atomiques (1, 2, 3, 4 et 5). Le modèle de plus haut niveau, qui contient tous les autres modèles est le modèle couplé *A*. Le second modèle couplé *B* est composé des modèles atomiques (3, 4, 5). La figure 5 présente le modèle couplé *A* avec 2 entrées "*IN*" et une sortie "*OUT*". Il contient 2 modèles atomiques MA1 et MA2 et un modèle couplé *B*.

La cohérence et la conservation des propriétés du système entre ces niveaux de hiérarchie sont résumées par la propriété de "fermeture sous composition". En effet dans le formalisme DEVS, chaque modèle est indépendant et peut être considéré comme une entité à part entière ou comme le modèle d'un système plus grand. Il a été montré dans [*ZEIGLER, 1984*] que le formalisme DEVS est fermé sous composition, c'est à dire que pour chaque modèle couplé DEVS, représenté par le couplage d'un ensemble de sous modèles, il est possible de construire un modèle atomique DEVS équivalent.

4. Simulation DEVS

La simulation est un procédé informatique visant à faire évoluer un système afin de prédire son comportement. Etablir une simulation exige donc la définition précise du comportement ainsi que la description des interactions qui existent entre les modèles.

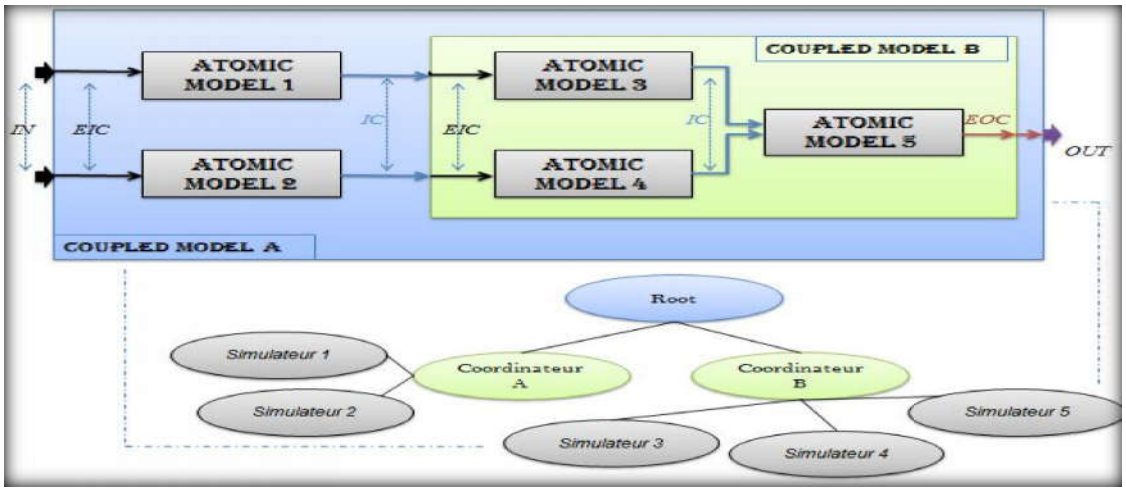

Figure 6: Arbre de classe du simulateur DEVS

L'une des propriétés importantes du formalisme DEVS est qu'il fournit automatiquement un processeur pour chacun des modèles. Le formalisme DEVS établit une distinction entre la modélisation et la simulation (figure 6) d'un système tel que n'importe quel modèle DEVS puisse être simulé sans qu'il soit nécessaire d'implémenter un processeur spécifique.

La figure 6 montre comment sont organisés les processeurs (*Root, coordinateur, simulateur*). Chaque modèle atomique est associé à un *Simulateur* chargé de gérer le comportement du modèle, et chaque modèle couplé est associé à un *Coordinateur* chargé de la synchronisation temporelle des modèles sous jacents. L'ensemble de ces modèles est géré par un processeur spécifique appelé *Root* [*ZEIGLER, 1984*].

5. Variantes de DEVS

Nous pouvons intégrer dans le multi-formalisme DEVS de nombreux autres formalismes ou méthodes de modélisation, nous trouvons *Parallèle-DEVS*, *Fuzzy-DEVS*, *Stocastic-DEVS* etc. Mais nous s'intéressons à la variante Parallèle-DEVS parce qu'elle s'adapte à notre travail, principalement les comportements des cyber-attaques.

5.1. Parallèle DEVS :

Cette variante a été développée par B.P Zeigler, elle a pour rôle de donner une technique simple de parallélisme des calculs. Le DEVS utilise la technique de hiérarchie des modèles qui sont repartis sur des calculateurs. Donc, il faut un point de synchronisation qui est nécessaire pour effectuer un test sur la causalité des événements.

Les chercheurs –dans [*CHOW, 1994*]- constatent que la clé pour répondre à ces exigences est de traiter correctement les collisions, c'est à dire le comportement lorsqu'un composant reçoit les événements externes en même temps que sa transition interne

préprogrammés. Les solutions précédentes tentent de définir le comportement de collision implicitement soit par la fonction de sélection ou en imposant la priorité d'une transition interne sur une transition externe collision. En revanche, la structure du formalisme parallèle DEVS proposé ici permet au modélisateur de définir explicitement le comportement de collision en utilisant la fonction dite de transition confluent δ_{con}

5.1.1. Le modèle atomique

$$M =< X, S, Y, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, ta > \quad (4)$$

- X : est l'ensemble de l'événement entré ;
- S:est l'ensemble d'état séquentiel ;
- Y: est l'ensemble de l'événement sorti ;
- $\delta_{int} : S \to S$ la fonction de transition interne ;
- $\delta_{ext} Q \times X \to S$ la fonction de transition externe,

  X^b Est un ensemble de sacs sur les éléments de X

  (S, e, ∅) = (S,e) où $Q = \{(s, e) | s \in S, 0 \leq e \leq ta(s)\}$

  e est le temps écoulé depuis la dernière transition d'état ;
- $\delta_{con}\ S \times X^b \to S$ la fonction de transition confluente ;
- $\lambda : S \to Y^b$ la fonction de sortie ;
- $ta : S \to R_{0 \to \infty}$ La fonction de l'avance de l'heure.

Dans le formalisme Parallèle DEVS, le modeleur est explicitement activé pour fournir la fonction de transition supplémentaire confluente qui capture le comportement de collision. La fonction de transition confluente réduit le calcul nécessaire pour des transitions d'état et le calcul d'ordonnancement en associant l'ensemble des transitions internes et externes quand elles se produisent en même temps [*CHOW, 1996*].

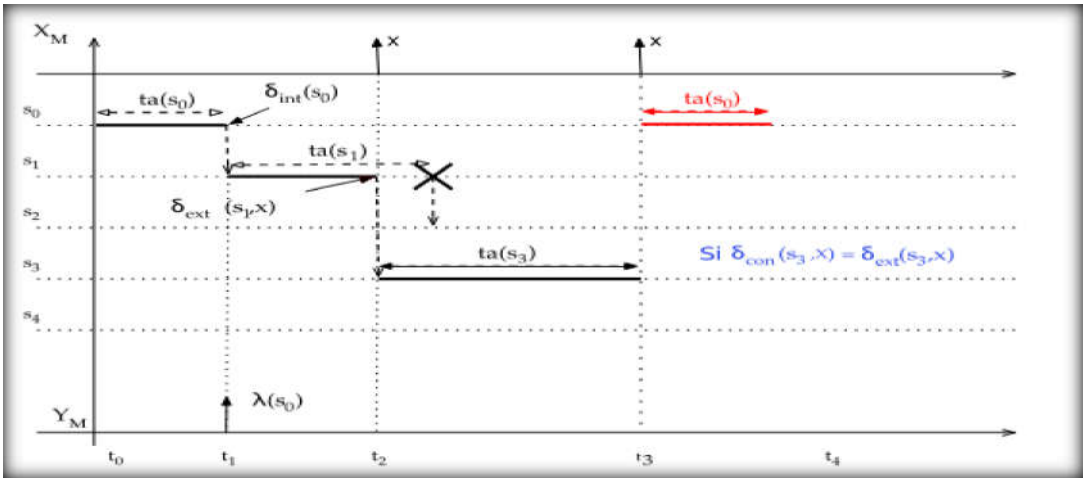

Figure 7: les transitions d'états en Parallèle DEVS

δ_{con} est la différence avec les formalismes de DEVS précédentes. Il donne le contrôle complet modélisateur sur le comportement de collision quand un composant reçoit les événements au moment de sa transition interne, *e* = 0 ou e = ta (*s*). Le Parallèle DEVS laisse cette décision de cette sérialisation à utiliser, le cas échéant, à la modélisation. La sémantique des parallèles DEVS sont illustrés dans la figure 7. Les transitions internes sont effectuées au moment de la prochaine épreuve pour tous les composants imminents ne recevant pas

d'événements externes. En outre, les événements externes générés par ces imminents déclenchent des transitions externes au non-imminents réceptifs (les composants pour lesquels il n'existe pas de transitions internes prévues lors de l'événement du temps de réception).
Toutefois, les composants pour lesquels les transitions internes et externes entrent en collision, la fonction de transition confluente est employée au lieu de deux fonctions de transition interne ou externe pour déterminer le nouvel état.
Dans la figure 7, toutes les transitions d'état sont les mêmes que celles dans le formalisme DEVS d'origine, sauf à t_2 et t_5 où les collisions se produisent. Lors de ces cas, les nouveaux états sont définis comme $S_3 = \delta_{con}(S_2, X_2)$ et $S_6 = \delta_{con}(S_5, X_5)$.

5.1.2. Le modèle couplé

La structure du modèle couplé révisée $DN =< X, Y, D, \{M_i\}, \{I_i\}, \{Z_{i,j}\} >$ *(5)*

- X: un ensemble d'événements d'entrée ;
- Y: un ensemble d'événements de sortie ;
- D: un ensemble de composants pour chaque i dans D_i ;
- M_i est une composante ;
  pour chaque i dans D U {self} ; I_i *est influencé de i* ;
  pour chaque j dans I_i, $Z_{i,j}$;

6. Conclusion

Nous avons présenté dans ce chapitre le formalisme DEVS développé par B.P Zeigler. Ce formalisme peut être défini comme une méthodologie universelle et générale qui fournit des outils pour modéliser et simuler des systèmes dont le comportement repose sur la notion d'évènements, et permet la spécification de systèmes complexes à évènements discrets sous forme modulaire et hiérarchique. Nous avons présenté une variante basée sur DEVS permettant de prendre en compte des imperfections sur les paramètres des modèles. L'utilisation du Parallèle DEVS est là pour corriger correctement les collisions.

Dans le chapitre suivant nous présenterons une nouvelle proposition de taxonomie des cyber-attaques et aussi une proposition d'un modèle général pour la sécurité des réseaux.

1. Introduction

Avec la multiplication et la complexité croissante des cyber-attaques, la sécurité informatique a besoin d'être améliorée par la structuration, l'organisation et de la classification des cyber-attaques existantes. En effet, il est impossible de générer tous les cas possible en tenant compte de toutes les attaques connues et inconnues. Pour régler ce problème, nous proposons une nouvelle classification des cyber-attaques. Dans la majorité des cas des classifications existantes, aucune d'elles n'est reconnue comme un standard, ni utilisée à grande échelle.

2. **Analyse des classifications existantes** :

Scientifiquement, le classement des signatures dans le domaine de l'informatique ou dans des autres sciences exige différents types de systèmes de classification selon le système étudié. Dans la littérature, il existe plusieurs travaux qui ont touchés la classification des cyber-attaques. Certains de ces travaux, viennent d'examiner les cyber-attaques en liste, ce qu'il appelle taxonomie en une seule dimension tandis que d'autres sont allés en profondeur dans les caractéristiques d'attaques, en développant de véritables taxonomies en multiple dimensions.

2.1. **Les taxonomies en une seule dimension**:

Parker et Neumann :

Les chercheurs décrivent une série de classes sur les abus informatiques environs de 3000 cas depuis vingt ans de travail. Ce travail a été amélioré par Neumann en neuf classes par rapport au premier travail [*HACHEM, 2014*]. L'inconvénient de cette taxonomie est que les types d'attaques sont moins intuitifs et plus difficile de se rappeler pour les trois simples types de menaces pour les fondements de la sécurité (Confidentialité, Intégrité et Disponibilité) dans une simple catégorie de menaces mais depuis l'existence des listes complexes d'attaques basées sur les événements (relation cause – effet) il est difficile de constater sa pertinence.

Brinkly et Schell :

Ils ont fournis une liste des abus informatiques et leurs techniques sans fournir une véritable taxonomie. Ils ont considéré que certaines classes correspondent à la menace sur les ordinateurs eux mêmes, tandis que les autres classes correspondent sur les menaces qui pèsent sur l'information traitée par les ordinateurs [*HACHEM, 2014*].

Cohen :

Il fournit un schéma de classification pour les aider dans l'évaluation de la sécurité ; la liste de travail contient 93 attaques sans être regroupées dans des catégories supérieurs [*HACHEM, 2014*]. Cette liste ne comprend pas seulement les attaques informatiques, mais aussi des incidents tels que la panne de courant et les intempéries. L'inconvénient principal de cette

liste est qu'il ne reste pas statique et doit être constamment en date pour le garder pertinent et pour couvrir toutes les attaques.

Koch et all :

Ils ont classé et répertorié dix classes de menaces basées sur des rapports techniques de plusieurs entreprises de sécurité [*HACHEM, 2014*]. Leur classement reste incomplet et aborde des types de menaces limités, telles que l'ingénierie sociale et le Cloud computing.

2.2. Les taxonomies de multi-dimension :

Hansman et Hunt :

Ils proposent une taxonomie qui se compose de quatre dimensions ; le vecteur d'attaque, la cible de l'attaque, les vulnérabilités et les charges utiles [*HUNT, 2005*]. Son utilité est démontrée sur un certain nombre des attaques bien connues. C'est la première taxonomie consacré aux attaques combinés et elle n'est pas complète.

Mishra et Saini :

Ils ont développés une méthode de classement de cyber-attaques à l'aide de la métrique d'une caractéristique et l'approche de la théorie de jeu pour classer les attaques sur les catégories les plus proches [*MISHRA, 2009*]. Leur classification est considérée comme une taxonomie en raison de la démesure dans les paramètres utilisés ; l'objectif de l'attaque, propagation de l'attaque, la vulnérabilité exploitée, la méthode utilisée, les actifs mal utilisés et leur effet sur l'actif. L'inconvénient de cette taxonomie est que les auteurs n'ont pas déterminés les métriques standards pour chaque catégorie pour classer les attaques.

Harrison et White :

Ils ont présentés une taxonomie qui considère la motivation, la méthodologie et les effets des événements de cyber qui peuvent affecter les communautés [*HARRISON, 2011*]. Cette taxonomie a utilisé deux vecteurs : le vecteur d'événements qui représente l'attaque et le vecteur d'effet qui reflète l'impact de l'attaque. Dans chacun des vecteurs cités au dessus, les auteurs utilisent plusieurs caractéristiques pour classer les cyber-attaques dans la taxonomie.

Uma et Padmavathi

Les auteurs ont donnés une classification des cyber-attaques [*UMA, 2013*]. L'étude tente de classer les attaques en fonction de diverses caractéristiques telles que la gravité, l'objet, de la légalité, afin de fournir une compréhension de la motivation à l'origine de ces attaques qui peuvent permettre aux programmeurs de développer des dispositifs de sécurité et les mécanismes basés sur le mode de l'attaque.

Simmons et all :

Ils ont proposés une taxonomie pour les cyber-attaques appelée AVOIDIT afin d'aider à l'identification et à la défense contre les attaques [*SIMMONS, 2014*]. Ils utilisent cinq grandes

classes pour caractériser la nature de l'attaque ; le vecteur d'attaque, l'impact de l'exploitation, défense, l'impact d'information et la cible. Dans leur travail, ils utilisent largement la notion de la vulnérabilité mais elle est omise dans leur taxonomie et la classe de défense reste abstraite.

3. Caractéristiques de la taxonomie

Après avoir examiné les taxonomies existantes, il est important de définir ce qu'une bonne taxonomie s'inclue des caractéristiques. Une bonne taxonomie doit satisfaire à plusieurs exigences pour une acceptation universelle. Un certain nombre d'exigence pour notre taxonomie est cité en dessous ;

- *Acceptée/ bien structurée* : la taxonomie doit être structurée de telle sorte qu'elle peut être approuvée ou fondée sur des travaux antérieurs généralement approuvés.

- *Compréhensible* : une taxonomie sera facilement compréhensible à la fois pour les spécialistes dans le domaine et avec un simple intérêt.

- *Sans Ambigüité* ; la classification doit être sans ambigüité. Si une signature ou un attribut appartient à une catégorie, il devrait être clair.

- *Conformité des terminologies* : les terminologies utilisées dans la taxonomie doivent être conforme avec les terminologies de domaine de la sécurité afin d'éviter la confusion et de bâtir sur des connaissances antérieures.

- *Exclusivité mutuelle* : ça veut dire que les catégories ne se chevauchent pas et que chaque attribut correspond à une catégorie au plus.

- Utilité : la taxonomie devrait être utilisable dans l'industrie de la sécurité et surtout par les équipes de domaine.

4. **Proposition d'une nouvelle classification** :

Dans le domaine de la sécurité des réseaux, il y a plusieurs travaux qui se concentrent sur les cyber-attaques. Certains de ces travaux visent les taxonomies. Dans la sécurité informatique, nous pouvons avoir trois catégories générales des taxonomies: les taxonomies des vulnérabilités, les taxonomies des attaques et les taxonomies pour la détection d'intrusion [*HARRISON, 2011*]. Les taxonomies de vulnérabilité cherchent à classer les vulnérabilités qui sont des trous de boucle du logiciel, en analysant les caractéristiques telles que leurs défauts associés, les méthodes d'introduction, effets de l'exploitation qui sont identifiées par les attaquants et les exploitées. Les taxonomies des attaques se concentrent sur le vecteur d'attaque utilisé pour attaquer un système, ainsi les différents types d'attaque ou méthodes utilisées pour compromettre un système, les vulnérabilités exploitées et les effets potentiels ou les résultats de l'attaque sur un système. Les taxonomies des systèmes de détection

d'intrusion classent les menaces par les mesures de détection d'intrusion et de signatures requises pour la détection.

Pour avoir une meilleure classification de cyber-attaques, nous se basons sur les attributs et les caractéristiques que nous aurons identifiées auparavant et respectées, en éliminant celles qui sont ambigües ou qui ne sont pas pertinentes pour une cyber-attaque et pour que la solution de défense soit réalisable et claire. Habituellement, lors d'une classification, nous utilisons de multiples catégories, chaque catégorie est utilisée pour classer au moins un attribut d'une seule signature.

La définition d'une taxonomie devrait passer, en réalité, par une identification des principaux objectifs à respecter. En effet, il est plus intéressant de dire que la sécurité est faible ou robuste vis-à-vis de la détection de tel ou tel type d'attaque. Au contraire, lorsque nous exprimons les résultats en distinguant chaque cyber-attaque prise individuellement (et non de manière générique à travers les classes d'attaques). Il faut constater que chacune des classifications existantes a été développée dans un but particulier (par exemple, comprendre les vulnérabilités pour renforcer les mesures correctives et défensives, appréhender les processus d'attaque ainsi que le comportement des attaquants, etc.). Il en résulte que les attributs identifiés dans une étude ne sont pas forcément pertinents pour une autre ayant un objectif différent [DAOUD, 2015].

Nous trouvons que dans certains travaux existants souffrent d'un manque de clarté dans la distinction entre les attributs et entre les attaques. Par exemple, certaines classifications regroupent le débordement de tampon et le déni de service sous le même attribut ; ce choix nous semble abusif car une attaque qui exploite un débordement de tampon peut causer un déni de service.

Notre objectif est de proposer une nouvelle proposition d'une taxonomie pour les cyber-attaques et que cette taxonomie n'aborde pas un type spécifique mais elle examine, en général, plusieurs types d'attaques. Le but de ce travail est de faire une étude approfondie de ces attaques dans le but de créer une prise de conscience sur les différentes catégories de notre taxonomie proposée et tout ça pour avoir des mesures de défense qui peuvent être engagées à l'encontre les attaques. Puisque les cyber-attaques sont un danger sur le comportement des individus et les réseaux informatiques, il est important de les classifier en catégorie pour faciliter la compréhension de ces menaces. Une taxonomie descriptive permet de comprendre les effets potentiels et la portée de ces attaques. Une telle taxonomie aidera à la modélisation et à la simulation correcte des attaques [DAOUD, 2015].

La proposition de notre classification est basée sur la vision d'un attaquant qui porte atteinte aux propriétés de la sécurité (confidentialité, intégrité et de la disponibilité). Cette classification a une petite particularité, elle ressemble à une séquence d'actions, d'où l'existence d'une certaine chronologie, dans le processus de l'attaque, entre ces dimensions. Cette taxonomie exprime fortement les propriétés dynamiques de l'attaque pour que nous puisons évaluer ces attaques pour la construction des défenses. Par ailleurs nous écartons une dimension de « *l'ingénierie sociale* » dans l'attribut type d'une attaque, elle ne sert pas à définir une catégorisation claire de la cybernétique.

Après avoir écarté les dimensions qui ne sont pas pertinentes dans notre classification et nous avons adapté les caractéristiques citées ci dessus qui sont seulement utiles. Nous montrons la proposition de la classification dans la figure 8. Nous citons les attributs les plus intéressants :

4.1. Le vecteur d'attaque (technique)

C'est la voie d'accès qui est donnée à un pirate pour obtenir un accès non autorisé à un hôte. Cette définition inclut les vulnérabilités, il peut exister plusieurs vulnérabilités pour réaliser une attaque réussite ou par tout moyen par lequel une attaque exploite pour mener une attaque.

- *Mauvaise configuration* : Un attaquant peut employer la configuration par défaut d'une application pour avoir un accès à un réseau ou un ordinateur pour causer une variété d'attaques. Les paramètres par défauts sont une cible facile pour exploiter une attaque.
- *Défaut de noyau* : Un attaquant peut utiliser une faille du noyau d'un système d'exploitation pour obtenir des privilèges pour l'exploitation des vulnérabilités du système d'exploitation.
- *Débordement du tampon* : Il est causé quand un morceau de code ne vérifie pas correctement la longueur d'entrée et que la valeur d'entrée n'est pas de la taille du programme attendu. Une attaque peut exploiter une vulnérabilité de type débordement du tampon qui conduit à une éventuelle exploitation de l'exécution du code arbitraire.
- *Validation d'authentification insuffisante* : Un programme ne parvient pas à valider l'authentification d'une application et/ou l'utilisateur envoyé au programme d'un utilisateur. un attaquant capte les informations d'identification utilisateur pour emprunter l'identité d'un utilisateur valide. C'est ce qu'on appelle la mauvaise permission.
- *Vulnérabilité.* Il s'agit d'une faiblesse de sécurité qui peut être de nature logique ou physique. Une vulnérabilité peut découler, par exemple, d'une erreur d'implémentation dans le développement d'une application, erreur susceptible d'être exploitée pour nuire à l'application (pénétration, refus de service, etc.). Elle peut également provenir d'une mauvaise configuration. Elle peut enfin avoir pour origine une insuffisance de moyens de protection des biens critiques, comme l'utilisation de flux non chiffrés, l'absence de protection par filtrage de paquets, etc.

4.2. Le type (Méthodes)

C'est-à-dire le moyen le plus utilisé par l'attaquant pour arriver à ses fins, comme les virus, les vers, Spam etc. Les types d'attaques sont bien détaillés dans le chapitre 1 dans la partie les types de l'attaque.

4.3. La cible de l'attaque

C'est-à-dire quelle est la destination visée par exemple du système d'exploitation, du réseau, etc.

- *Application* : C'est d'attaquer un logiciel spécifique.
- *Système d'exploitation* : C'est d'attaquer une vulnérabilité au sein du système d'exploitation particulier

- *Réseau* : C'est de cibler un réseau particulier ou d'accéder à travers une vulnérabilité à un réseau.
- *Local* : C'est de cibler un ordinateur local.

4.4. Les résultats (l'impact de l'information)

C'est-à-dire qu'une fois une attaque a eu lieu, donc elle a un impact sensible sur l'information de diverses manières telles que la divulgation, perturbation, découverte, etc.

- *Déformation* : c'est qu'une attaque provoque des modifications sur un fichier. C'est-à-dire les données du fichier de la victime.
- *Perturbation* : c'est qu'une attaque consiste à perturber sur des accès que ce soit un changement d'accès ou la suppression de l'accès utilisés par la victime.
- *Destruction* : c'est qu'une attaque cause une suppression des fichiers ou une information de la victime. Son impact est plus malveillant.
- *Divulgation* : c'est qu'une attaque offre un affichage des informations pour les personnes non autorisés.
- *Découverte* : c'est qu'une attaque utilise des informations pour lancer une attaque sur une cible particulière.

Et par cette classification proposée des cybers-attaque, nous pouvons affirmer la vision d'un attaquant pour porter atteinte aux propriétés de la sécurité par la formule suivante :

- $\exists\ (Un\ Vecteur\ d'attaque \wedge Type\ d'attaque \wedge Une\ Cible) \Rightarrow Un\ Résultat\ d'une\ attaque$

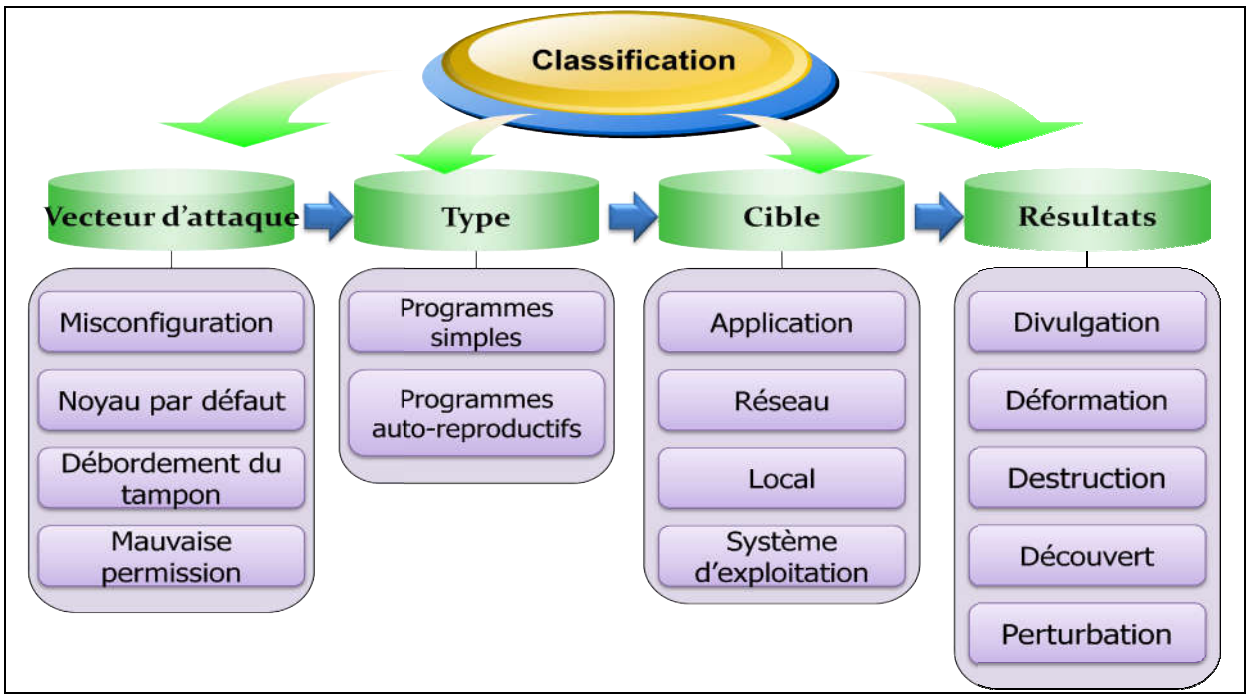

Figure 8: Classification proposée pour les cyber-attaques

Cette taxonomie sert à décrire le comportement d'une attaque selon la compromission de l'intrus à un système. La description de cette synthèse de taxonomie facilite une meilleure représentation de ces attaques dans la simulation.

Certains travaux précédents cités -dans la section analyse des taxonomies existantes pour les cyber-attaques- montrent la création d'un espace de noms communs pour toutes les vulnérabilités et les exploits. Les taxonomies d'attaques ne parviennent pas à exprimer formellement leurs propriétés dynamiques. Pour cela nous devons présenter un modèle général pour la sécurité des réseaux.

5. Modèle Général pour la sécurité des réseaux

Dans cette partie, nous traitons l'utilisation de la structure d'entités de systèmes pour spécifier les modèles hiérarchiques et de les organiser pour la réutilisation d'un modèle d'archive de base [DAOUD, 2015].

Le formalisme de la structure de l'entité du système est un système de représentation des connaissances structurelles qui organisent systématiquement une famille de structures possibles d'un système. Une telle famille caractérise la décomposition, le couplage et les relations entre les entités en matière de taxonomie. Une entité représente un objet du monde réel. La décomposition d'une entité concerne la façon dont il peut être décomposé en sous entités. Les spécifications de couplage racontent comment les sous entités peuvent être couplés en ensemble pour reconstituer l'entité. La relation de taxonomie concerne les variantes admissibles d'une entité.

Il existe trois types de nœuds dans l'arbre, ces types de nœud, nous pouvons les voir dans la figure 9 de la structure de l'entité du système de cyber-attaques. Un nœud d'entité représente un objet du monde réel. Il existe deux types d'entités, à savoir l'entité composite et entité atomique. Une entité composite est définie en termes de d'autres entités qui peuvent être soit atomique ou composite, tandis que l'entité atomique ne peut pas décomposée en sous entités. Chaque entité peut avoir des variables attachées. Elle peut aussi avoir plusieurs aspect / spécialisation [*CHI, 2001*] [*ZEIGLER, 2000*] [*GABRIEL, 2010*].

- (1) Le nœud d'aspect *Atomic-dec* est relié par une ligne verticale à partir de l'entité composite *ATOMIQUE* (Atomic), il représente une décomposition de l'entité en deux sous-entités *COMPOSAN*T (Components) et *SECURITE* (Security) et que ces sous-entités sont distinctes.
- (2) Le nœud de spécialisation *Net-spec* est relié par une double ligne verticale à partir de l'entité du nœud racine *RESAEAU* (Network). Il définit une taxonomie de l'entité et il représente la façon dont l'entité peut être classée en spécialisation.
- (3) on peut avoir aussi, une entité multiple qui est reliée par triple lignes verticales, c'est l'exemple de *RESEAUX* (Networks)

Après avoir présenté les trois types des nœuds du formalisme SES (Structure de l'entité du système) pour les cyber-attaques, nous essayons de décrire toutes les entités dans la figure 9:

Au début, nous trouvons le nœud racine de la structure de l'infrastructure informatique nommée *RESEAU* (Network). Comme nous savons tous, nous pouvons avoir deux types de réseaux : des réseaux qui se composent de plusieurs réseaux et d'un seul réseau atomique. Tout ça est représenté par la spécialisation (*Net-spec*) de l'entité *RESEAU* en deux sous-entités *ATOMIQUE* et *COMPOSITE*. L'atomicité et la composition sont concernées par l'architecture du réseau et non celle du formalisme [DAOUD, 2015].

Au début, nous commençons par l'entité *ATOMIQUE*, chaque réseau atomique est constitué par des *COMPOSANTS* (Component) et en parallèle par une *SECURITE* (Security) pour ces derniers par le nœud aspect (*Atomic-dec*). Comme nous sachons tous chaque *COMPOSANT* informatique, nous le trouvons en *MATERIEL* (Hardware) et en *LOGICIEL* (Software) par le nœud aspect (*Component-dec*). Concernant l'entité *MATERIEL* se décompose en plusieurs entités (*BORDURE, INFRASTRUCURE, BUREAU*, etc.) par l'entité d'aspect (*Hardware-dec*) et chacune de ces dernières classent certains entités, nous citons à titre d'exemple l'entité *BORDURE* (Routeur, Pare-feu, etc) par l'entité de spécialisation (*Bord-spec*). La même chose pour le nœud *LOGICIEL* qui se décompose en deux sous-entités *OS* (Système d'exploitation) et les *SERVICES* (Services) par l'entité d'aspect (Soft-dec) et chacun de ces deux entités classent certains entités atomiques (Windows, Linux, Solaris).

Parallèlement, nous reviendront au nœud *SECURITE* qui est à son tour composé en deux sous-entités sous le nom *CYBER-ATTAQUE* qui génère des scenarios d'attaque et *CYBER-SECURITE* qui analyse et contre ces cyber-attaques. Pour réaliser une cyber-attaque, il y a un processus ou une méthodologie qui sera suivi par les pirates en commençant par l'entité de spécialisation *FURTIVITE* (Stealth). Après la furtivité, l'attaquant a deux spécialisations la *RECUPERATION* (Recovry) des données ou bien *ECOUTE PASSIVE* (Passive Listining). Pour la récupération des données ou des informations à partir d'une cible, il y a plusieurs techniques qui sont classées par l'entité de spécialisation (*recovry-spe*), les techniques classées sont *REPLICATION, INJECTION, AUTOREPRODUCTION et DESTRUCTION*.

Parallèlement à la cyber-attaque, il y a la *CYBER-SECURITE* qui se compose de plusieurs outils de protection qui sont : *ANALYSEUR, IDS, IPS ANTI-VIRUS* etc.

Une fois, nous avons terminé la description du nœud *ATOMIQUE*, nous reviendrons maintenant au nœud *COMPOSITE*. Chaque réseau est divisé en plusieurs niveaux détaillés tels que les *RESEAUX* (Networks), un multiple d'entité qui peut lier plusieurs groupes de réseaux et la *LIAISON* (Link) les relie tous. Nous trouvons qu'il y a une classification de la liaison des réseaux, un est *FILAIRE* (Wired) et l'autre *SANS-FIL* (Wireless) qui sont spécialisées par l'entité (*link-spec*). Dans l'entité *FILAIRE,* il y a plusieurs topologies de liaisons dont nous citons : *BUS, RING, STAR et HUB* [DAOUD, 2015].

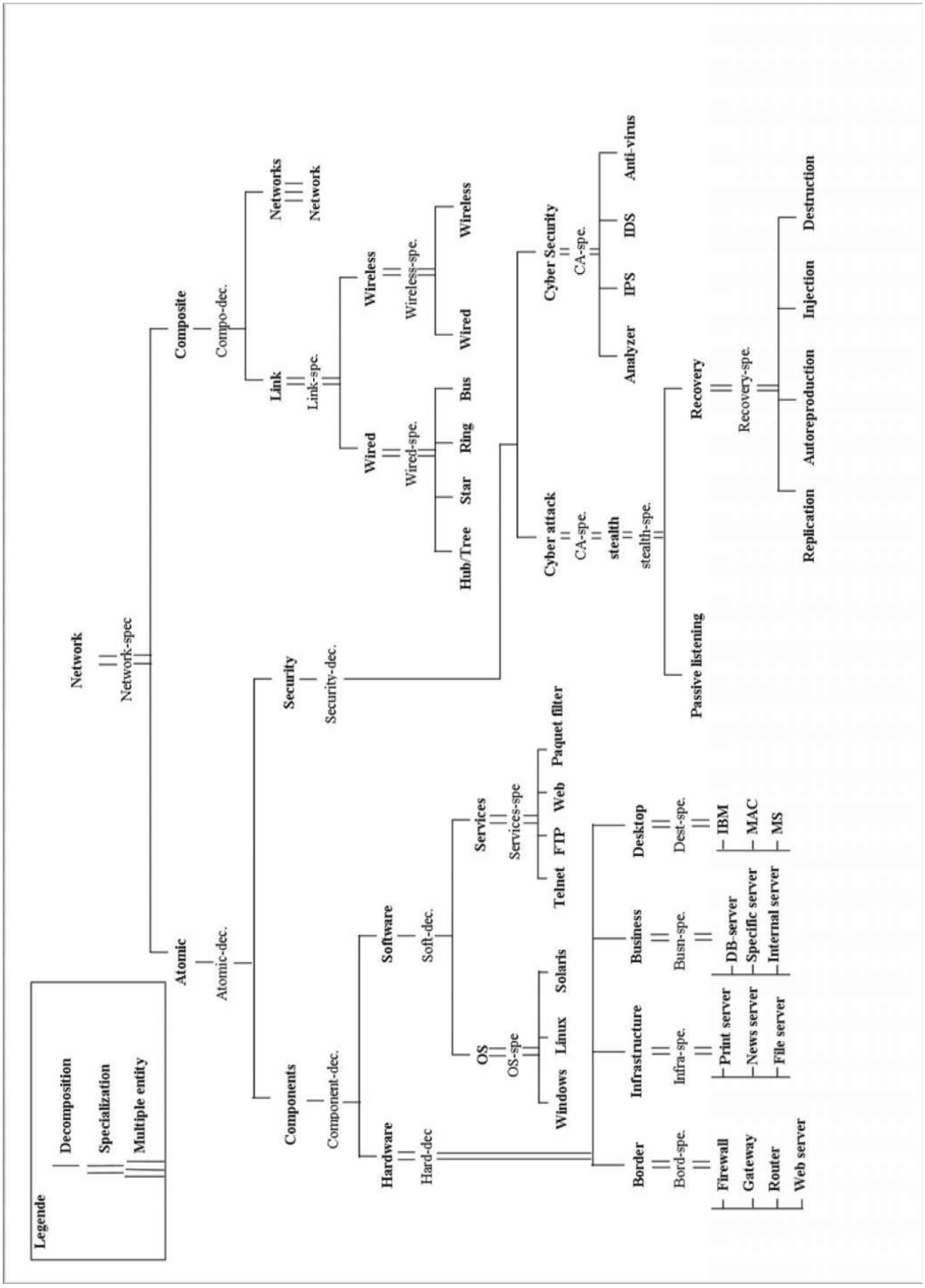

Figure 9: Le modèle général pour les réseaux informatiques

6. Conclusion

Dans ce chapitre, nous avons proposé une nouvelle méthodologie pour classer les cyber-attaques. Cette taxonomie repose sur les aspects dynamiques d'une attaque et que cette taxonomie repose sur la relation de causalité entre les différentes catégories et nous avons proposé un modèle général de base pour la structure des réseaux qui est de conception hiérarchique et modulaire.

Maintenant, nous nous focaliserons dans le chapitre suivant sur l'implémentation de ce que nous avons proposé par le formalisme DEVS pour modéliser et simuler les cyber-attaques.

1. Introduction

Ce chapitre présente un exemple d'une cyber- attaque qui permet d'attaquer une cible dans un réseau dont le vecteur d'attaque est le débordement du tampon et donné aussi un aperçu sur le modèle développé par le formalisme DEVS de l'attaque de débordement du tampon, en suite, nous simulons par l'outil DEVS-Suite développé par l'équipe du centre Arizona pour la modélisation intégrative et de Simulation. Notre but est de valider le modèle proposé par l'utilisation de l'outil DEVS-Suite.

2. Description de l'attaque

Dans notre cas, l'attaque se réalise entre deux machines virtuelles, la première machine comporte un système d'exploitation Linux (Ubuntu 14.0.0) et la deuxième machine comporte le système d'exploitation Windows XP pack 3. Dans cette machine, nous installons le serveur de capacité FTP et le logiciel immunité debugger.

Cette partie de travail décrit un cas d'une attaque d'un client qui est une victime d'une attaque de vulnérabilité FTP (File Transfert Protocol). Nous utilisons notre ordinateur pour échanger des informations entre les deux systèmes. Nous utilisons le serveur de capacité (Ability server), c'est un outil qui offre le nécessaire de fonctionnalités puisqu'il intègre Web, dispose des capacités de transferts et de courrier électronique. Nous avons pensé qu'il satisfait nos besoins et met en œuvre des efforts puissants pour notre attaque. Bien que l'utilisateur fasse une diligence raisonnable, nous sommes heurté par un exploit par un débordement du tampon. L'application installée est vulnérable à un débordement du tampon STOR. Le serveur de capacité FTP est vulnérable au « stockage d'un fichier » et « l'ajout à un fichier » en utilisant les commandes du protocole FTP : STOR et APPE. Le serveur de capacité encapsule un serveur FTP, compatible avec le protocole FTP bien établi. Ce protocole est construit autour du modèle client/serveur comme c'est généralement vu dans les applications TCP/IP. Le protocole FTP prend en charge le transport de fichiers d'un système à l'autre. Dans des situations normales, un compte doit être rempli sur le coté de serveur afin d'obtenir l'accès. Un serveur FTP anonyme est une variante selon laquelle aucune information de compte ne doit être fournie. Le client accède au serveur via un compte « login anonyme » par lequel la vérification du mot de passe n'est pas activée et il suffit de fournir une adresse électronique comme mot de passe ou sans mot de passe à tous.

Connexion de contrôle : elle est constituée d'une façon classique de client/serveur. Au démarrage, le serveur FTP ouvre le numéro de port TCP du FTP bien connu 21 et attend une connexion client. Lorsque le client effectue une ouverture active sur le port pour établir la connexion, une boite de dialogue est démarrée avec le serveur afin de présenter des informations spécifiques *UserId et Mot de passe* de la session. La connexion de contrôle reste

active pendant un cycle de vie complet de la session FTP et est utilisée pour transférer les commandes et réponses vers les deux machines.

Connexion de donnée : elle est créée chaque fois qu'il y a un échange de données entre le client et le serveur. Il est important que le transfert soit finalisé une fois le client termine la connexion de données.

2.1. Le débordement de la pile

Le débordement de la pile se produit généralement quand un tampon est remplacé. Un tampon a une limite définie en performance de mémoire du même type. L'existence de certains langages tel que C, assembleur etc. n'ont pas une prise en charge intégrée pour le contrôle du frontière. Lorsque le tampon est alloué, c'est la responsabilité du programmeur de s'assurer qu'il utilise la mémoire tampon correctement et qu'il n'écrit pas en dehors de la mémoire tampon.

3. Le Processus de l'attaque :

3.1. Reconnaissance :

Une fois l'installation du serveur a été faite, donc, nous procédons à la recherche des informations sur les serveurs FTP sur des sites d'Internet. Dans notre reconnaissance, il est probable que nous trouvons des serveurs FTP vulnérable qui sont nécessaire dans notre cas d'étude. La particularité du serveur de capacité FTP, c'est qu'une chaine de caractère est toujours envoyée au cours de la première connexion au serveur. Cette chaine est fixe et ne peut pas être modifiée par un paramètre de configuration.

3.2. Numérisation :

Une fois, nous avons identifié la cible possible, nous pouvons vérifier si la cible exécute ce logiciel vulnérable. Cela peut être fait par une variété d'outils tels que le Nmap : Il y a une possibilité de lancer la commande « *nmap* » sur le port FTP pour le balayage. La commande *nmap* sera exécutée pour vérifier que ce port est ouvert par cette celle-ci :

C:\Documents and Settings\Amine> **nmap -P0 -p21 192.168.198.132**

Après l'exécution de la commande ci dessus, elle signale que le port TCP 21 sur ce serveur est ouvert. La commande nmap permis le balayage complet des réseaux. Une autre option intéressante est la prise d'empreintes TCP/IP. Avec cette option, la commande « nmap » tente de déterminer quel système d'exploitation pendant que le serveur est en cours d'exécution. Ceci est possible car le protocole TCP documente seulement les combinaisons valides dans les options TCP. Puisque chaque système d'exploitation plus ou moins réagit uniquement aux options non valides spécifiées dans un paquet spécial conçu.

3.3. Accès au système :

A ce stade, Nous écrivons un script avec le langage Ruby qui produit un débordement du tampon. Ce script vise à construire à ce qu'un exploit se réalise sur la machine de linux. Ensuite, nous démarrons le serveur de capacité FTP et nous attachons le processus du serveur dans le programme de débogueur de l'immunité, cela nous permet de voir ce qui se passe à l'application quand on lance le script Ruby contre l'application serveur FTP. Une fois nous exécutons le script, l'application s'écrase et nous aurons le résultat de l'accident dans le débogueur.

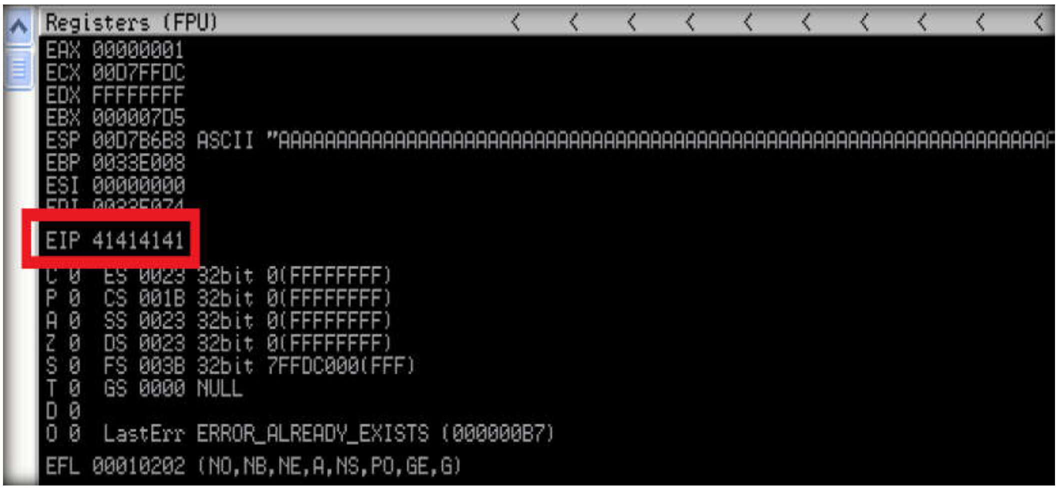

Figure 10: étape 01 réception du script

<u>Détermination de l'adresse EIP et ESP</u>

La figure 10 affiche les informations de l'accident dans le débogueur tel que : écrasement de registre EIP (l'adresse de retour) par notre tampon et que l'adresse de retour contrôle le flux d'exécution pour l'application.

Si nous pouvons contrôler le registre EIP (adresse de retour), nous pouvons avoir l'écrasement de l'application, pour cela nous exécutons un code qui peut se placer minutieusement en mémoire qui pointe vers un registre ESP d'un autre tampon au moment de l'accident. Pour cela, Nous avons besoin de déterminer le point dans notre tampon de 2000 octets qui écrase le registre EIP et qui sera pointé vers le registre ESP. Pour la réalisation de cette étape, Nous avons besoin de deux outils qui font partie du cadre du Metasploit, les outils sont *pattern_create.rb* et *pattern_offset.rb*. L'utilisation du script pattern_create.rb pour générer une nouvelle mémoire tampon de 2000 octets pour une chaine de caractères uniques. Une fois que nous aurons la chaine de caractères, nous utilisons pattern_offset.rb pour déterminer la position du registre EIP récupéré par le débogueur.

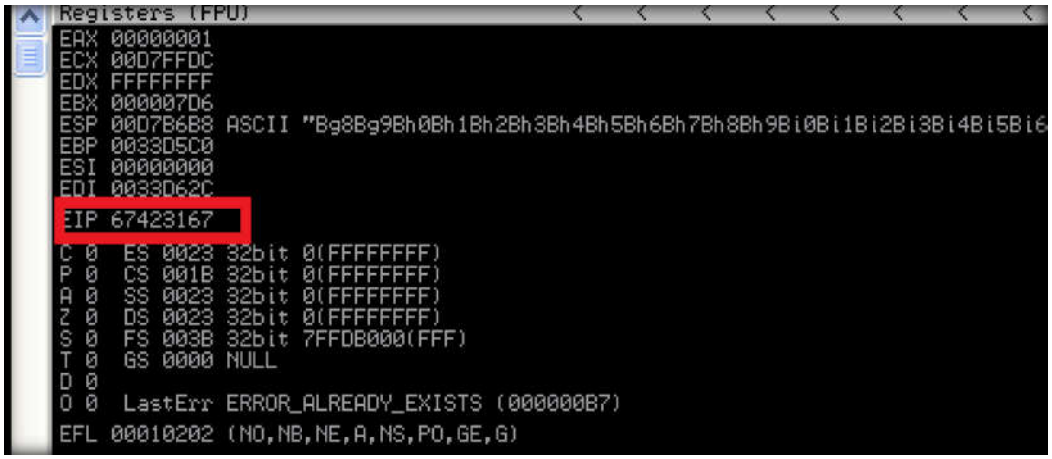

Figure 11: Etape 02 détermination de @ de retour par Metasploit

Durant cette étape, nous avons modifié le tampon existant en remplaçant la chaine de caractères et qui nous montera que le script va bloquer à nouveau l'application. Ce résultat est représenté en figure 11.

Association d'EIP à ESP ou (Ecrasement)

Maintenant, que nous connaissons les positions dans notre tampon qui correspondent aux deux registres, nous pouvons créer une nouvelle mémoire tampon qui va remplacer le registre EIP et qui va pointer vers le registre ESP dans lequel nous pouvons placer notre Shell code qui va être exécuté par l'application. C'est un moyen de détourner le flux d'exécution de l'application à exécuter le code Shell, c'est qu'on a pu coder en forçant l'adresse mémoire par un écrasement du registre EIP et en pointant vers l'emplacement de l'ESP au moment de l'accident dans la mémoire tampon. Cette adresse pourrait changer si l'application redémarre ou pourrait être différente sur d'autres machines. Nous utilisons plus précisément la commande JMP ESP qui doit être prélevée d'une application du système, il sera plus fiable car les adresses mémoires de l'application ne sont jamais changées entre les différentes versions du système d'exploitation, pour identifier la commande JMP ESP, nous allons utiliser le débogueur pour récupérer tous les processus chargés en mémoire par la commande JMP ESP. La figure 12 montre cette association des registres.

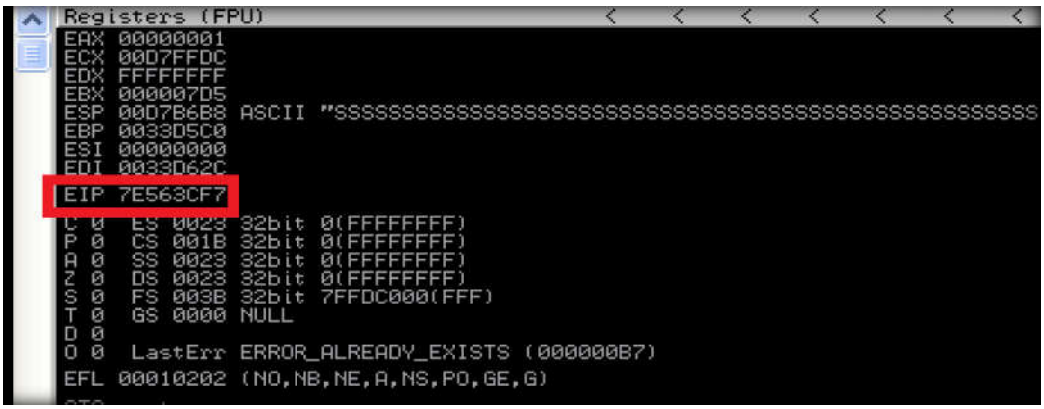

Figure 12: Etape 03 Association de @ ESP au EIP

Parmi ces fichiers, nous utilisons le fichier user21.dll. Nous allons ensuite modifier le tampon et donc la commande JMP ESP effacera le registre EIP. Après l'exécution du script d'exploit contre l'application cible, nous déterminons si la commande a été exécutée avec succès, détournée le registre EIP et pointa vers l'emplacement mémoire où nous injecterons notre code que nous voulons exécuter. Cela signifie que nous pouvons inclure notre code dans le tampon que nous allons l'envoyer et qui sera exécuté par l'application. Avant de créer le Shell code, nous devons déterminer combien de places en mémoire, nous avons pour notre Shell code en utilisant le débogueur. Pour cela, nous cliquons sur le registre ESP au moment de l'accident qui contient le reste de notre tampon et nous découvrons l'emplacement dans la mémoire. Nous pouvons utiliser le *Metasploit* pour la création des Shell code.

Injection et Génération

Une fois que le Shell code a été généré par le Metasploit, nous insérons l'exploit en modifiant le tampon, nous ajoutons une séquence de non-opérations (leur codage en assembleur est x90) qui n'a aucune commande de fonctionnement et cela agira comme rembourrage de Shell code à décoder. Ce Shell code a une coquille de liaison inverse, nous activons l'outil d'écoute des connexion des réseaux entrant sur le port 5555. L'exécution de Shell code envoie un succès d'un TCP inverse lié à l'ordinateur distant à l'ordinateur de l'attaquant sur le port 5555 qui permet d'interagir avec l'interface de commande distante de l'ordinateur cible.

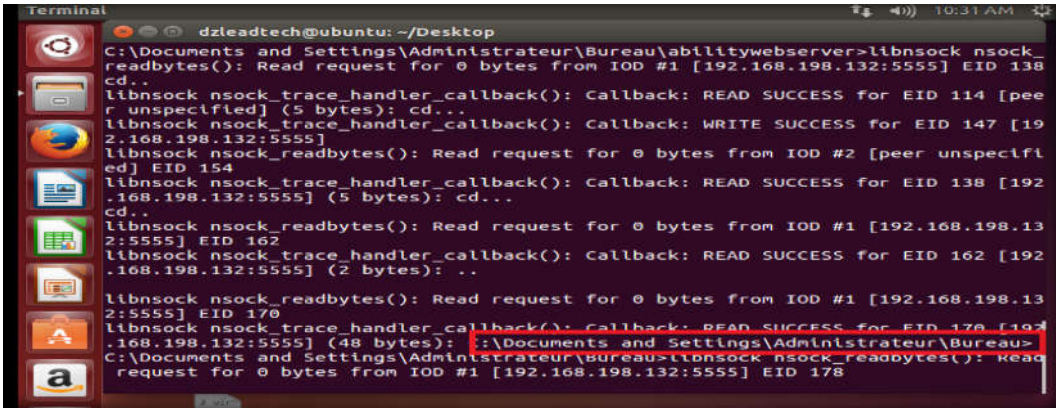

Figure 13: Etape 04 Exploit

Maintenant, nous avons identifié dans la figure 13 avec succès une vulnérabilité de débordement du tampon dans l'application du serveur de capacité FTP avec la commande STOR et ensuite, nous avons construit un exploit du travail pour attaquer l'application via une vulnérabilité identifié pour obtenir l'exécution de code à distance sur l'ordinateur.

3.4. Maintien de l'accès :

L'attaquant a eu accès au poste de travail de la victime. Dans le cas où le serveur de capacité sera exécuté sous le compte administrateur, l'accès à la machine pouvant être gardé en lançant la commande « net user » du DOS. Un pirate informatique intelligent utiliserait une politique correcte, pour ne pas être détecté comme suspect. Une utilisation finale plus prudente choisirait d'exécuter le serveur sous un compte avec nos privilèges.

4. Modélisation

Comme nous l'avons vu, le formalisme DEVS est basé, pour la modélisation, sur deux types de modèles : les modèles couplés et atomiques. Ces modèles possèdent des ports d'entrée, des ports de sortie et des variables d'état. Chaque modèle communique grâce à l'envoi et à la réception de plusieurs types de messages. Chaque message génère des évènements qui sont stockés dans un échéancier, qui est une structure de données composée d'évènements classés suivant un ordre chronologique, la tête de l'échéancier représentant le futur immédiat, et la queue le futur plus lointain. La simulation consiste à faire évoluer les états des modèles dans le temps en fonction d'évènements.

Un modèle est donc une représentation simplifiée du comportement observable et de la structure d'un système réel. Le but de la modélisation est de construire un modèle afin de résoudre un problème d'analyse ou de conception. Et aussi de faire la description d'un système de manière vérifiable, compréhensible et réutilisable. La résolution de ces problèmes, appliquée sur le modèle, permet d'obtenir une solution dans le monde du modèle. Le but final de ce modèle est d'effectuer des expérimentations sur ce type d'attaque -décrit ci dessus pour les cyber-attaques. Le modèle est développé par notre proposition de notre équipe. La figure 14 présente le modèle implémenté dans l'outil Suite-DEVS.

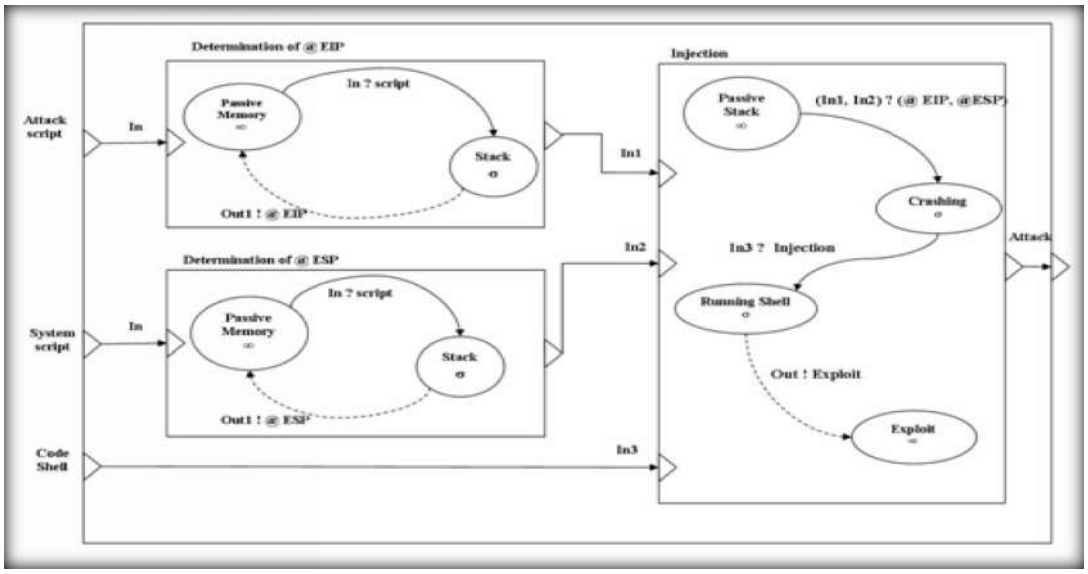

Figure 14: modèle couplé de l'attaque de débordement du tampon

Le modèle général est composé de trois modèles atomiques comme décrit par la figure 14.

Détermination de l'adresse EIP : c'est un modèle couplé pour récupérer l'adresse de retour de l'application du serveur de capacité FTP.

Détermination de @ de retour = (X, Y, S, δ_{int}, δ_{ext}, λ, t_a)

```
X={(In, A*2000)} // fichier 01
```

Y={(Out, EIP)}
S={mémoire, pile, @EIP} // {pile, buffer01, @EIP}
δ_{int}(pile)=@EIP
δ_{ext}(memoire, e, In ?A*2000)= pile
λ(pile)=(Out !@EIP)
t_a(pile)=20
t_a(memoire)=infini

Détermination de l'adresse ESP : c'est pour récupérer l'adresse du registre ESP du tampon pour écraser l'adresse de retour du tampon du fichier.

Détermination de @ de ESP = (X, Y, S, δ_{int}, δ_{ext}, λ, t_a)

X={(In,buffer02)} //fichier 02
Y={(Out, ESP)}
S={mémoire, pile, @}
δ_{int}(pile)=@ESP
δ_{ext}(memoire, e, In ?buffer02)= pile
λ(pile)=(Out !@ESP)
t_a(pile)=20
t_a(memoire)=infini

Injection : c'est la phase pour injecter le Shell code choisi pour réaliser l'attaque.

Injection du shellcode = (X, Y, S, δ_{int}, δ_{ext}, λ, t_a)

X={(In,EIPnew)} //fichier 02
Y={(Out,Attaque (0/1))}
S={pile, écrasement, injection, nouvelleEIP, taille}
δ_{int}(écrasement)= nouvelleEIP
δ_{ext}(pile, e, In ?EIP, ESP)= écrasement
λ(écrasement)=(Out !NEIP)
t_a(écrasement)=20
t_a(pile)=infini
δ_{int}(injection)= taille
δ_{ext}(nouvelleEIP, e, In ?shellcode)=injection
λ(injection)=(Out !attaque)
t_a(injection)=50
t_a(nouvelleEIP)=infini

5. Simulation

Le but de la simulation est de voir où va nous emmené un modèle à partir d'un état initial et de conditions initiales. Construire un modèle à partir d'un système réel et simuler son comportement devrait alors mener aux mêmes résultats que ceux d'une expérimentation réelle puis d'observer et codifier les résultats expérimentaux [*ZEIGLER, 1984*]. La réalisation d'une simulation dans un domaine donné demande de nombreuses connaissances et compétences. Puisque nous sommes dans le domaine des cyber-attaques, les cyber-attaques sont très complexe, leur modélisation est d'obtenir des résultats plus proches de la réalité, la simulation est utilisée pour anticiper les réactions et les comportements des cyber-attaques modélisées.

Comme notre travail repose sur une représentation à haut niveau, nous nous focalisons sur les changements d'état suite à des événements. En effet, les modifications des modèles ont lieu suite à des événements, telle que la détermination de l'adresse de retour et d'injecter le code. De plus, les événements ne sont pas programmés à intervalle régulier. Suivant ces caractéristiques, nous concluons que la simulation à événements discrets convient tout à fait à notre cas d'étude, puisque les événements discrets permettent de ne pas réévaluer l'état du modèle lorsque cela n'est pas jugé nécessaire, c'est-à-dire lorsqu' aucun événement ne s'est produit. Dans le processus des cyber-attaques, les changements des états sont effectués par des événements. Le choix de la simulation à événement discret par rapport à la simulation continue est que la complexité de cette dernière grandit avec le nombre de paramètres et il devient rapidement impossible de modéliser les systèmes complexes de manière purement analytique. L'intérêt de la simulation à événement discret apparait lorsque le phénomène simulé utilise des échelles de temps différentes.

5.1. Simulateur de l'attaque de débordement

Pour mieux comprendre l'attaque, nous avons essayé de développer un simulateur construit par l'équipe. Ce simulateur a été conçu par le langage Java à l'aide de l'IDE Eclipse. Ce logiciel est construit par une fenêtre principale décrit dans la figure 15. Nous essayons d'expliquer la fenêtre, nous avons divisé en parties numérotées :

- *La section (A)* décrit les entrées de l'attaque qui sont au nombre de quatre entrées :
  - *Taille* : définit le nombre de caractères utilisés par l'application du serveur Ability TFP avant l'adresse de retour.
  - *N_EIP* : Cette entrée définit l'adresse de retour choisie par l'attaquant.
  - *Nop* : c'est le nombre de non opération avant le Shell code.
  - *Shell code* : c'est la case où on injecte le Shell code.
- *La section (B)* est décrite dans un tableau contenant la liste des processus qui sont exécutés en parallèle avec le serveur FTP.
- *La section (C)* comporte un bouton de simulation et un espace d'affichage
  - *Le bouton Simuler* : il permet de lancer la simulation de l'attaque par les entrées et le processus choisi.
  - *Espace d'affichage :* il affiche des messages d'avertissement une fois les champs d'entrées ne sont remplis par des valeurs.

- *La section (D)* affiche des résultats partiels pour chacune des étapes essentielles de l'attaque. Nous avons trois étapes Shell code valide, Ecrasement et Injection.
    - *Shell code Valide* : le Shell code est validé par l'exclusion des mauvais caractères.
    - *Ecrasement* : cette étape permet de comparer l'adresse de retour et l'adresse ESP du processus choisi.
    - *Injection* : c'est une étape qui permet de vérifier la taille du Shell code et l'espace restant du processus.
- *Section (E)* permet d'afficher le résultat final de la simulation si l'attaque a réussi ou non.

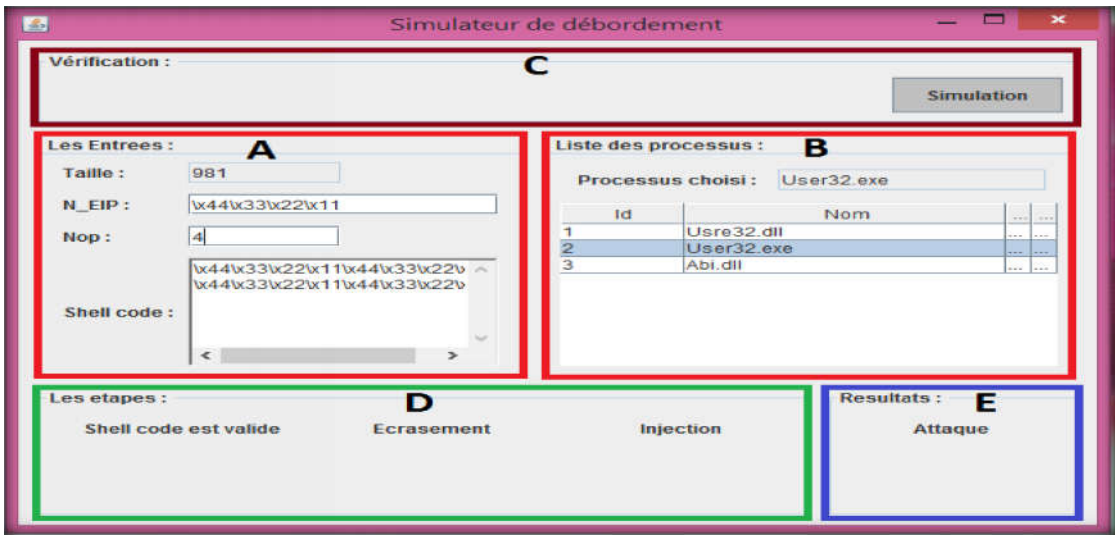

Figure 15: L'interface du simulateur de l'attaque de débordement

La figure 16 présente le succès de l'attaque et l'affichage des entrées valide de cette simulation pour enfin déterminer la structure du buffer. Dans ce cas toutes, les étapes sont en couleur verte ce qui veut dire que toutes les étapes sont vérifiées et validées et l'attaque a réussi.

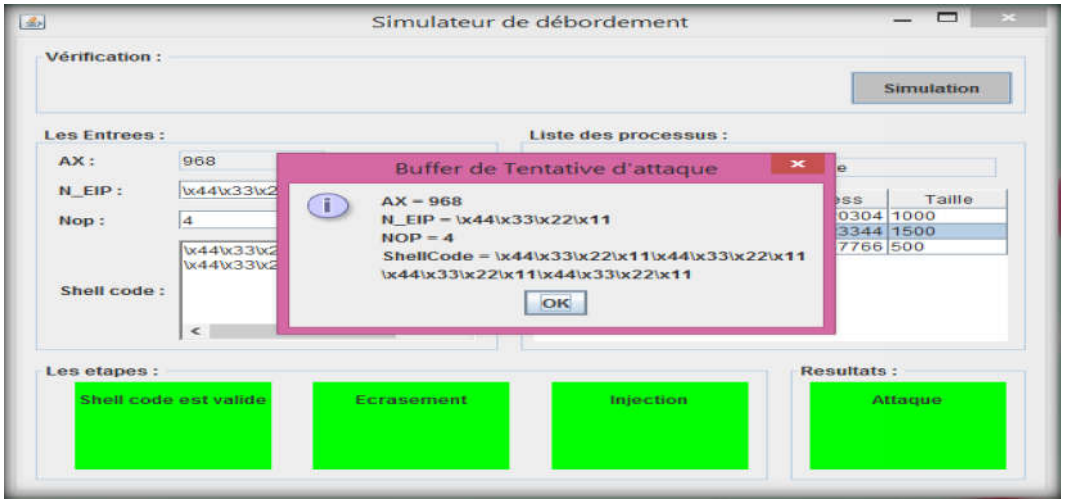

Figure 16: Un succès de la simulation de l'attaque

La figure 17 présente un résultat d'échec de la simulation de l'attaque par l'affichage de la couleur rouge pour les étapes qui ne sont pas vérifiées et la couleur verte pour l'étape vérifiée et que l'attaque est en rouge à son tour. Dans ce cas, nous trouvons que le Shell code n'est pas valide parce qu'il contient des mauvais caractères et que l'étape écrasement n'est pas vérifiée parce que l'adresse de retour (N_EIP) et l'adresse de (ESP) du processus choisi ne sont pas les mêmes.

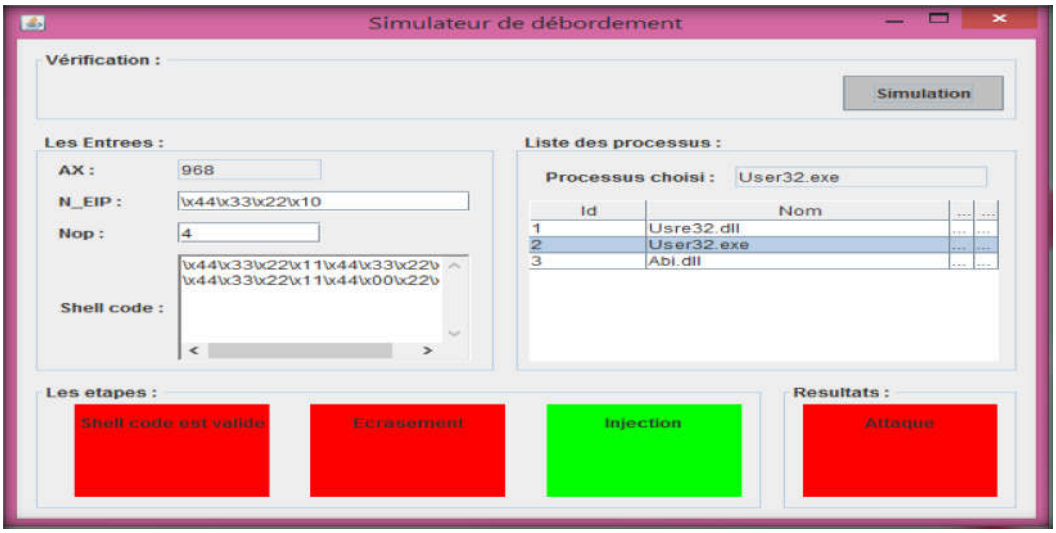

Figure 17: Un échec de la simulation de l'attaque

5.2. Simulateur Suite-DEVS

Il est hautement souhaitable pour des outils de simulation pour permettre aux utilisateurs d'observer la structure et le comportement d'un système réel. Certaines fonctions de base sont nécessaires dans un simulateur pour un utilisateur d'être capable d'interagir avec le système.

Suite-DEVS est une application basée sur Java utilisée pour représenter des modèles et de leurs interactions dans un environnement graphique et interactive. Suite-DEVS, qui est basé sur un système modulaire formel, cadre de modélisation hiérarchique, est parmi les outils disponibles aujourd'hui de simulation qui offre de multiples vues simultanées, de l'information de modèle [ERIC, 2009]. Il prend en charge la simulation de modèles décrits en fonction de la spécification du système à événements discrets (DEVS). Chaque modèle est visuellement représenté par l'une des deux formes de base: un rectangle rempli de composants de base, ou un contour rectangulaire pour les modèles hiérarchiques qui sont composés d'un ou plusieurs composants internes. Chaque composant peut avoir des accès d'entrée et de sortie, qui sont utilisés pour transmettre des messages entre les composants.

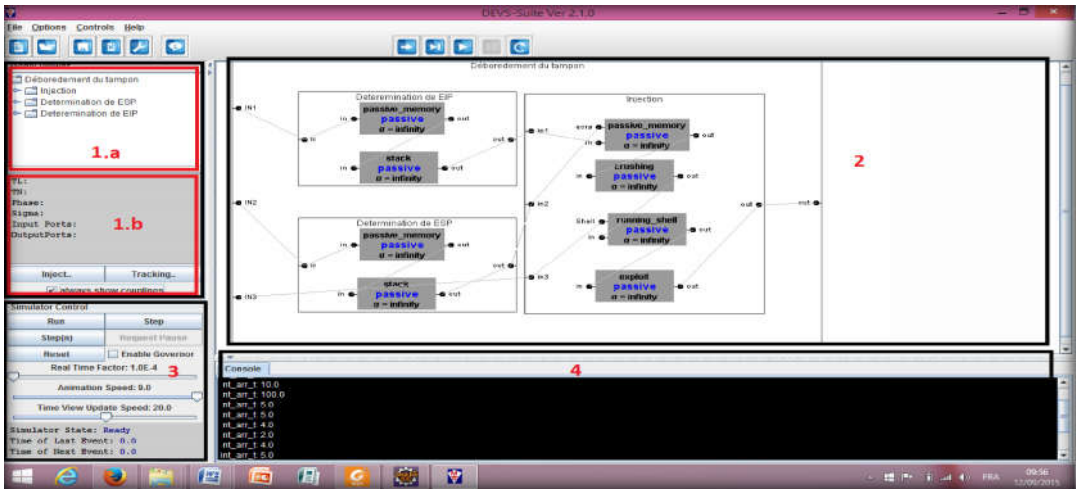

Figure 18: Le Suivi (Traking) de l'environnement Suite-DEVS

La figure 18 montre un exemple de l'interface du Suite-DEVS en simulant notre cas d'étude. Il y a 4 sections principales ; (1) l'afficheur de modèle (Visualiseur), (2) afficheur de la simulation, (3) le contrôle de simulateur et (4) la fenêtre du suivi.

Après le chargement d'un modèle, l'afficheur du modèle est rempli avec une liste des composants atomiques et couplés contenus dans ce modèle dans la partie (*1.a*). Immédiatement en dessous de la liste des composants, il existe une boite qui dresse la liste *(1.b)* des variables prédéfinies relatives au modèle sélectionné par l'utilisateur.

Dans notre modèle, nous pouvons voir que le composant *débordement du tampon* possède trois (03) ports d'entrées, et un port de sortie. Dans la boite, nous trouvons deux boutons Injecter et Suivi (Traking)

- *Injecter* : pour fournir manuellement les données à des moments arbitraires dans la simulation.
- *Suivi (Traking)* : c'est pour initialiser les fenêtres de visualisation de données.

L'afficheur de simulation (2) affiche le modèle visuellement, y compris les composants hiérarchiques. Notre modèle (*débordement du tampon*) contient trois (03) composant couplés et que chacun d'eux contient des composants atomiques. Les trois composants couplés sont appelés ; *Détermination de @ EIP, Détermination de @ ESP et Injection.*

En dessous de cette dernière fenêtre, il y a la fenêtre du Suivi (Traking) (4) qui contient la console de sortie.

Enfin, en bas à gauche (3) c'est le contrôle de simulation. De là, l'utilisateur peut contrôler les actions du simulateur. Dans cette fenêtre, il existe un certain nombre de boutons et curseurs pour l'utilisateur, nous citons leur fonctionnement en dessous ;

- *Facteur du temps réel* ; il contrôle à quelle vitesse le temps logique du programme progresse par rapport au temps de l'horloge. Cette variable peut ajuster l'échelle de temps de simulation logique afin d'obtenir une réponse en temps réel plus rapide ou

plus lente. Si ce nombre est fixé à 1, alors chaque unité de temps dans le programme va prendre une seconde pour passer.

- *Vitesse d'animation* ; il détermine la rapidité des messages qui vont être déplacés autour de l'écran entre les composants. La valeur vraie de curseur d'animation est comprise entre 1 et 9,9. Ces deux paramètres du programme sont soumis à des limitations matérielles et logicielles du système.
- *Temps d'exécution de simulation* : il est défini en termes de temps logique, qui est défini comme un nombre réel compris entre zéro et l'infini. Afin de réduire le traitement inutile, le programme calcule le prochain changement d'état dans la simulation et pointe vers elle, puisque rien de pertinent qui se passe entre ces changements d'état.

Prés des curseurs, il existe un groupe de boutons qui contrôle le comportement du simulateur :

- *Exécuter* : il exprime le simulateur à l'étape automatiquement grâce à l'interaction de modèle que l'utilisateur arrête l'exécution manuellement.
- *Etape* : il exécute le programme jusqu'à ce qu'un changement d'état se produit, puis retourne le contrôle à l'utilisateur.
- *Reset* : c'est réinitialiser les composants à leurs états initiaux. Cette option est disponible à tout moment sauf lorsque la simulation est en cours.
- Simulation :

Après la conception correcte du modèle de l'attaque de débordement du tampon, on charge le modèle à partir de la liste des modèles. Pour commencer la simulation et obtenir les résultats qui valident le travail. Nous commençons par le déroulement de la simulation étape par étape.

Dans notre modèle, il y a trois entrées, donc nous essayons injecter dans ces trois entrées des valeurs dans les endroits mentionnés dans la figure (19).

Pour les modèle atomiques *PASSIVE_MEMORY* pour chaque des modèles couplés *DETREMINATION DE @EIP* et *DETERMINATION DE @ ESP*, ils sont initialisés par leurs phases est *PASSIVE*, un sigma= *INTINITY* et une couleur grise. Dans notre cas, des modèles atomiques du modèles sont tous à l'état passive parce qu'aucun n'est reçu comme événement.

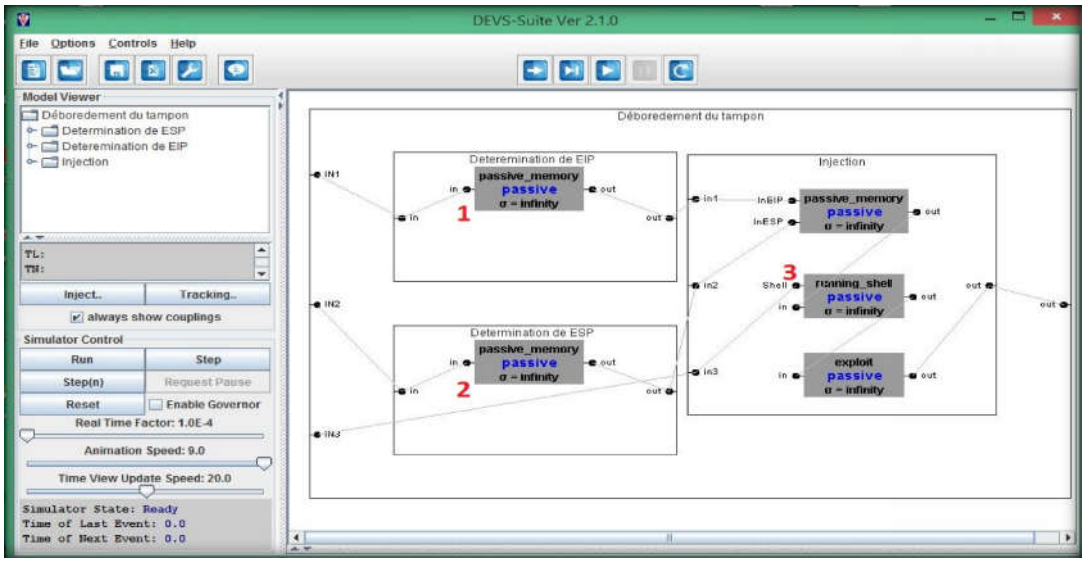

Figure 19: le modèle de l'attaque

Puis, nous injectons les valeurs dans les ports *IN, IN, SHELL* pour chacun des modèles respectivement *DETERMINATION @EIP, DETERMINATION @ESP et EXLOIT*. Dans ce cas, il y a un changement des états pour les modèles atomiques et que chaque modèle prend des nouvelles valeurs. A titre exemple, le modèle *PASSIVE_MEMORY* change sa phase à *ACTIVE* et le sigma =5 et sa couleur à *CYAN* ça veut dire que le modèle a reçu un événement extérieur. Ces changements sont observés dans la figure (20) en dessous.

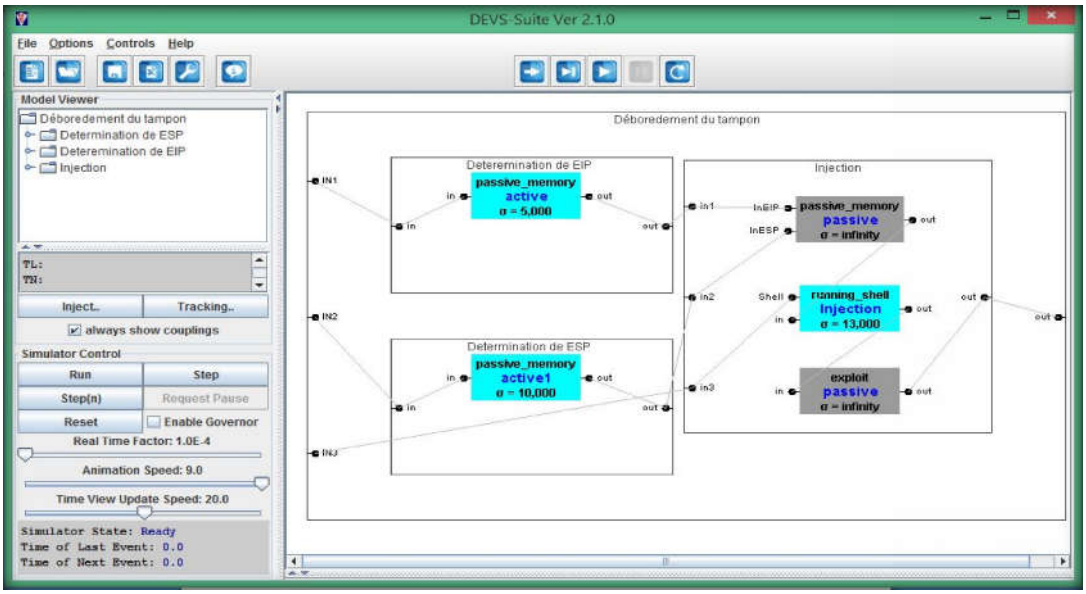

Figure 20 : Injection des entrées

Une fois, nous lançons la simulation, nous sommes sûr qu'il y a des changements et obtenir des résultats. Pour mieux comprendre la simulation et les résultats, nous simulons étape par étape pour bien voir les sorties et les entrées de chaque modèle atomique et aussi les changements des phases et les sigmas (*ta* dans le formalisme).

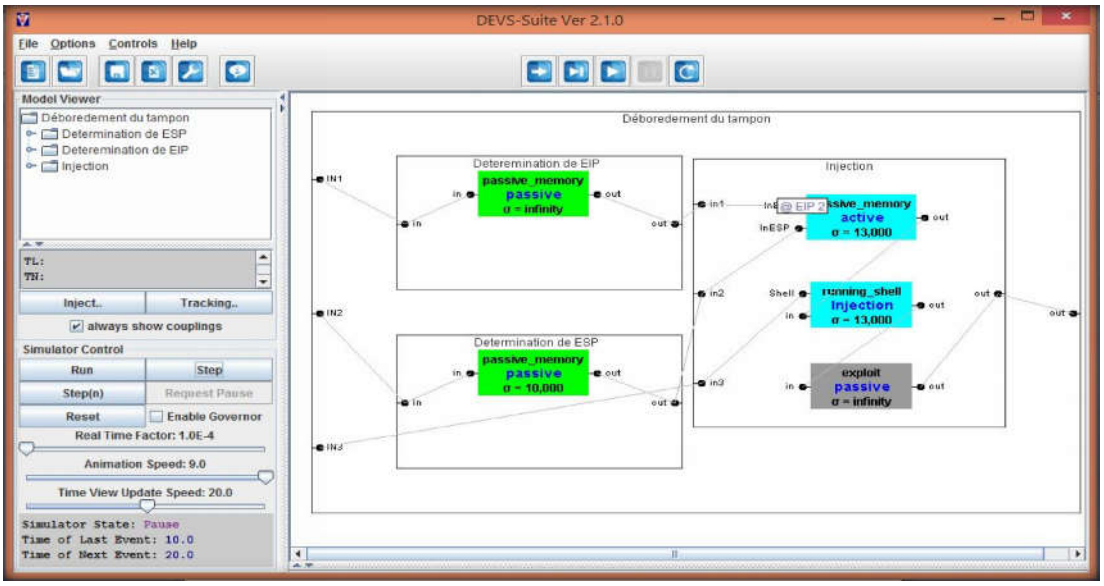

Figure 21 : Les changements les phases

A chaque fois, que nous continuons la simulation, il y a des changements au niveau des modèles atomiques dans leurs phases et leurs sigma que ce soit par la génération des évènements par la fonction interne, soit par la réception des événements extérieurs entre les modèles.

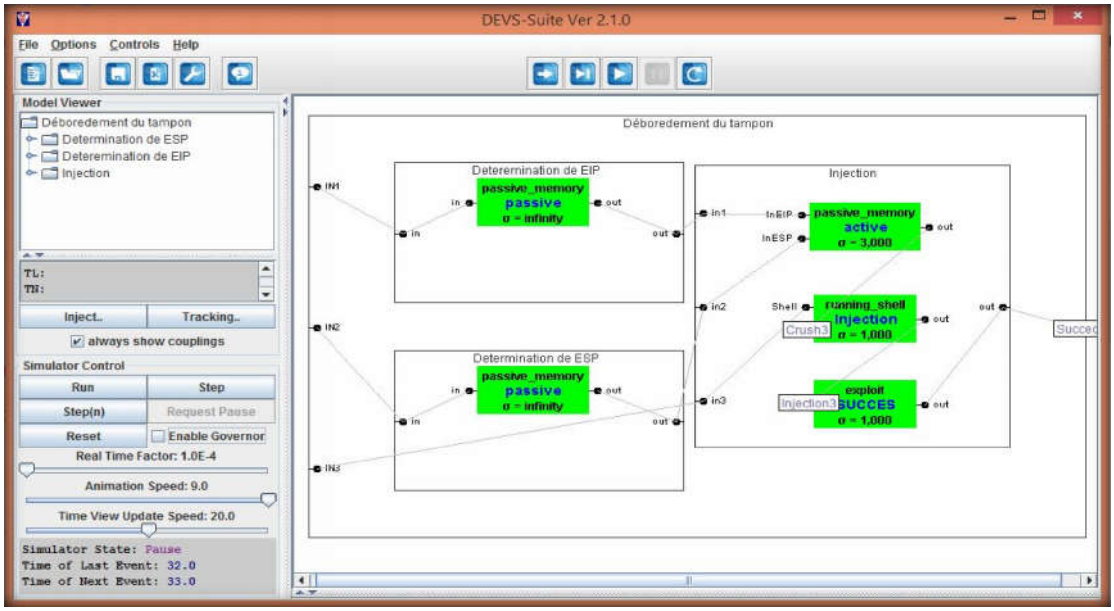

Figure 22: les résultats de la simulation

A la fin, de la simulation, nous obtenons un résultat dans la figure (22) qui récapitule que l'exploit est fait avec succès et qu'il y a des changements au niveau des modèles atomiques et pour que les résultats soient bien compréhensibles et les changements des phases soient visibles. Nous sollicitons les figures (23), (24), (25) en dessous pour les différents modèles atomiques.

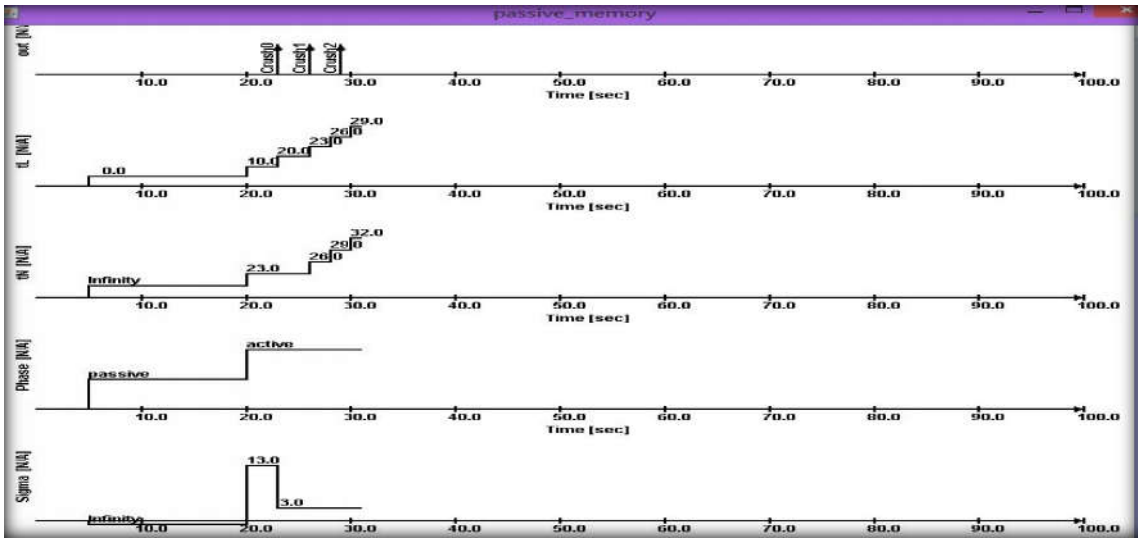

Figure 23: les changements du modèle PASSIVE_MEMORY

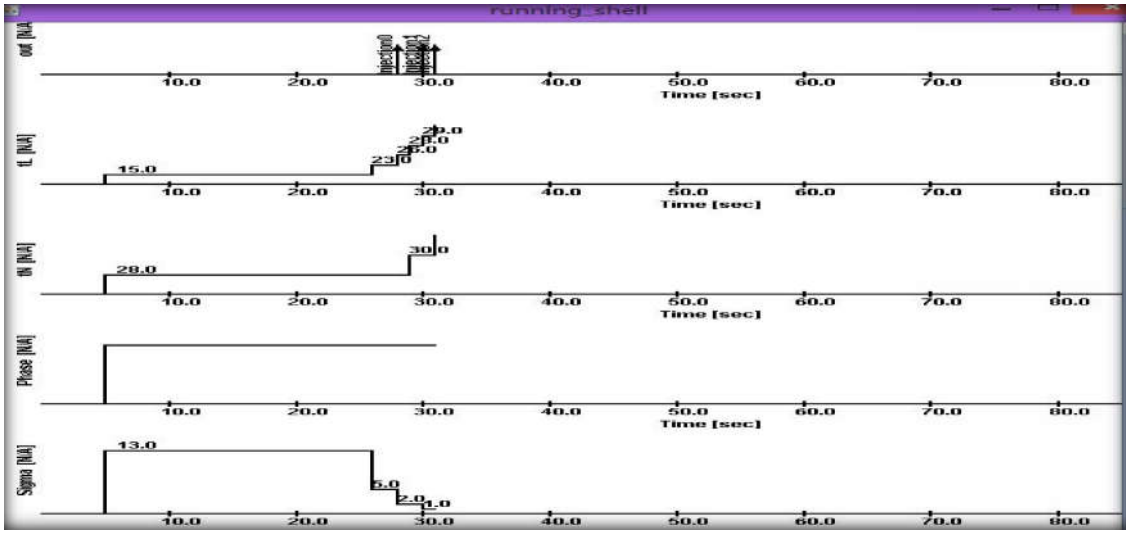

Figure 24: les changements du modèle RUNNING_SHELL

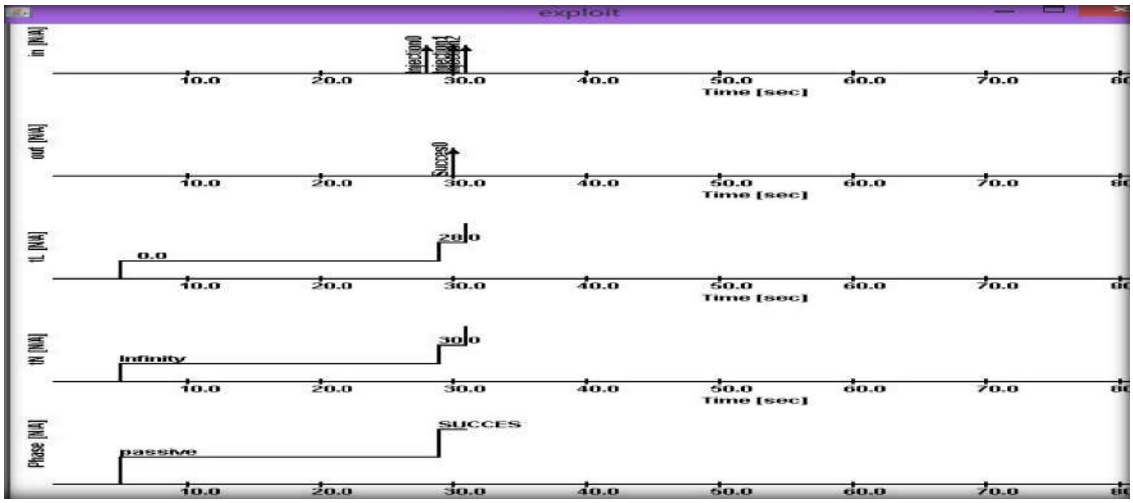

Figure 25: les changements du modèle EXPLOIT

Chacun des modèles reçoit des événements du modèle liés auparavant par des ports IN. Et à chaque réception d'un événement extérieur se déclenche un multiple de changements au niveau de phases (PASSIVE, ACTIVE, SUCCES etc.), sigma (le temps diminue à chaque exécution d'une fonction interne sans avoir un événement extérieur) et aussi TL Time last, TN (Time Next) TN et TN déterminent à un instant donné combien du temps a été écoulé et combien du temps sera pour exécuter la prochaine étape. À la fin, le modèle génère des sorties du modèle à partir de port OUT.

6. Conclusion

Dans ce chapitre, nous avons identifié une vulnérabilité de débordement du tampon dans l'application du serveur de capacité FTP avec la commande STOR et ensuite, nous avons construit un exploit du travail avec succès pour attaquer l'application via une vulnérabilité identifié pour obtenir l'exécution de code à distance sur l'ordinateur. Puis nous avons pu réaliser un petit simulateur concernant notre cas d'étude (débordement du tampon) pour mieux comprendre le fonctionnement de l'attaque. Enfin, nous avons illustré l'implémentation de notre approche à l'aide de l'outil DEVS-Suite, qui est un environnement de modélisation et de simulation développé par l'équipe du centre Arizona pour la modélisation intégrative et de Simulation. L'exemple traité dans notre étude est la modélisation et la simulation des cybers-attaques par le choix du débordement du tampon. Le but de ce modèle est valider ces systèmes complexes tel que les cyber-attaques et pour avoir au futur un meilleur produit pour la sécurité des réseaux .Nous avons présenté quelques résultats obtenus. Du fait de la complexité et de la taille des systèmes à étudier, il est improbable qu'une personne seule puisse traiter l'ensemble des problèmes et exploiter les résultats.

Conclusion Générale

De nos jours, les cyber-attaques sont à la fois beaucoup plus répandues et beaucoup plus complexes que celles des années précédentes, ce qui nous impose de maintenir la sécurité en réduisant le danger de ces attaques et que les systèmes informatiques doivent être protégés pour assurer les objectifs de sécurité (CIA) Confidentialité, Intégrité et la Disponibilité. Nous avons pour objectif à ce sujet de modéliser et de simuler des cyber-attaques par le formalisme DEVS.

Nous avons présenté dans le premier chapitre, un état de l'art sur les cyber-attaques et la cyber-sécurité et nous avons décrit les différents modèles utilisés pour la modélisation et la simulation des cyber-attaques. Puis, dans le deuxième chapitre le choix du modèle ou bien particulièrement, le formalisme DEVS par ces points forts pour la modélisation et la simulation dans le cas des cyber-attaques qui est connu par son expressivité et sa généralisation de modèles. Dans le troisième chapitre, nous avons développé notre travail par deux contributions. La première est de proposer une nouvelle taxonomie pragmatique pour les attaques et le deuxième est de proposer un modèle général pour les structures des réseaux qui est basé sur la structure de l'entité du système qui est de conception hiérarchique et modulaire. Enfin, nous avons présenté une implémentation d'une analyse des vulnérabilités de dépassement du tampon et modélisé cette attaque par le simulateur Suite-DEVS développé à l'université Arizona.

Parmi les points forts de ce travail est que le modèle général proposé s'adapte avec la modularité et l'hiérarchie du formalisme DEVS, ce qui confirme notre choix du modèle pour la modélisation. Notre taxonomie proposée répond au besoin de la classification des cyber-attaques et en englobe en général la majorité des taxonomies existantes.

En perspective nous proposons l'amélioration des résultats obtenus de la simulation en les comparants avec d'autres travaux. Ce travail et le développement d'une approche générale de modélisation pour les cyber-attaques et en plus nous proposons l'intégration des outils de machine d'apprentissage et DEVS pour la réduction de la recherche de la cyber-attaque dans leur ensemble, ceci pour la rapidité et la performance, ces facteurs sont très importants pour la prévention des intrusions d'où l'amélioration de la qualité des logiciels et plus généralement augmenter le niveau de la cyber-sécurité.

Bibliographie

[ANGLANI,2000]	A.Anglani, Caricato, A. Grieco and F. Nucci, "*EVALUATION OF CAPACITY EXPANSION BY MEANS OF FUZZY-DEVS",* 14th European Simulation Multiconference (ESM)- Simulation and Modelling, 2000, pp 128-133, Belgium, ISBN 1-56555-204-0
[*BARROS, 1995*]	F. Barros,"Dynamic *Dynamic structure discrete event system specification: a new formalism for dynamic structure modeling and simulation*". In Proceedings of Winter Simulation Conference, 1995
[*BHAR, 2011*]	M. Bhardwaj and G.P. Singh, "*Types of Hacking Attack and their Counter Measure*", *International Journal of Educational Planning & Administration*, Volume 1, Number 1, pp. 43-53, 2011.
[*CHI, 2001*]	S.D. Chi, J. S. Park, K.C. Jung and J. Lee, "*Network Security Modeling and Cyber Attack Simulation Methodology*", ACISP 2001, Springer, LNCS 2119, pp. 320-333, 2001
[*CHOW, 1996*]	A.C Chow, "*Parallel DEVS : A parallel, hierarchical, modular modeling formalism and its distributed simulator*", Computer Simulation International, Vol 13, Number 2, pp 55-67, 1996.
[*CLUSIF, 2005*]	Clusif, "*les virus informatiques*", Espace menace-groupe virus, Décembre 2005, France.
[DAOUD, 2015]	M. A. DAOUD, Y. DAHMANI, " *ARTT Taxonomy and Cyber-attack Framewok*", NTIC 2015, pp 32-37, IEEE, Mila, ALGERIA.
[*DEN, 2006*]	D. Denning,"A View of Cyberterrorism Five Years Later. Naval Post Graduate School, Monterey, Canada, 2006.
[*ELBO, 2012*]	A.ELBOUCHTI, A.HAQIQ," *Modeling Cyber-Attack for SCADA Systems Using CoPNet Approach*", IEEE, 2012.
[*ERIC, 2009*]	J. H. Eric, "*Design and analysis of view synchronization in devs-suite*", ARIZONA STATE UNIVERSITY, 2009.
[*FOVINO, 2009*]	I.Fovino, M. Masera, A.D. Cian, "*Integrating cyber attacks within fault trees*", Reliability Engineering and System Safety, Elsevier, 2009.
[*GABRIEL, 2010*]	A. Gabriel, Wainer, J. Mosterman, "*Discrete-Event Modeling and Simulation: Theory and Applications*", December 17, 2010 by CRC Press.
[*GER*]	J. Gerphagnon, M. Albuques, "*Attaques Informatique*", Rio de Janeiro Brazil.

[*HACHEM, 2014*]	N. Hachem, "*Technique de Mitigation des Cyber-Attaques basée sur MPLS",* L'université Pierre Et Marie Curie, France, Juillet 2014.
[*KNOWN, 1996*]	Y. Kwon, H. Park, S. Jung, and T. Kim, " *Fuzzy-DEVS Formalisme : Concepts, Realization and Application*." Proceedings AIS 1996, pages 227–234, 1996.
[*KIM, 2012*]	J.Y. Kim and H.J. Kim, "*A MODELING METHODOLOGY FOR CYBER-SECURITY SIMULATION*", Proceedings of the 2012 Winter Simulation Conference, IEEE, 2012.
[*LIU, 2013*]	Y.Liu, W.Gu "*An effective recognition method for network attack*", Optik, pp 4824-4828, Elsevier, 2013.
[*NOEL, 2008*]	S.Noel, S.Jajodia, "optimal ids sensor placement and alert prioritization using attack graphs", vol.16, pp 259–275, USA, Springer, 2008.
[*NTAIMO, 2002*]	L. Ntaimo, B.P. Zeigler, "*Expressing a forest cell model in parallel DEVS and timed cell-DEVS formalisms*". Proceedings of the 2004 Summer Computer Simulation Conference, 2002
[*PAN, 2012*]	S.Pan, T.Morris, U.Adhikari, V.Madani," *causal event graphs cyber-physical system intrusion detection system"*, CSIIRW'12, ACM, October 30- November 2, 2012, Oak Ridge, TN, USA, , 2012.
[*PAUL, 2014*]	S.Paul, R.Vignon-Davillier," *Unifying traditional risk assessment approaches with attack trees*" , journal of information security and applications, pp 165-181, Elsevier, 2014.
[*POOL, 2012*]	N.Poolsappasit, R.Dewri, I.Ray, "Dynamic Security Risk Management Using Bayesian Attack Graphs", transactions on dependable and secure computing, vol. 9, no. 1, pp 61-74, IEEE, 2012.
[*SHIN, 2014*]	J.Shin, H.Son, R.Khalilur, G.Heo, "*Development of a cyber security risk model using Bayesian networks*", Reliability Engineering and System Safety, pp 208-217 Elsevier, 2014.
[*SIMMONS, 2014*]	C. Simmons, C. Ellis, S. Shiva, D. Dasgupta, Q. Wu, "*AVOIDIT: A Cyber Attack Taxonomy*", University of Memphis.
[*TROC, 2003*]	A.Troccoli, G.Wainer, "*Implementing parallel cell-DEVS*", IEEE, Proceedings of the 36th Annual Simulation Symposium, 2003.
[*UHMA, 2001*]	A.Uhrmarcher, "*Dynamic Structures in Modeling and Simulation: A Reflective Approach*". ACM Transactions on Modeling and Computer Simulation, vol. 11 2001, pages 206–232, 2001.

[*UIT, 2006*]	Union internationale des télécommunications, " *Guide de la cyber sécurité pour les pays En développement* ", 2006
[*UMA, 2013*]	M. Uma, G. Padmavathi, "*A Survey on Various Cyber Attacks and their Classification*", International Journal of Network Security, Vol.15, No.6, PP.391-397, Nov. 2013.
[*WANG, 2012*]	C.Wang, N.Du, H.Yang, "*Generation and Analysis of Attack Graphs*", International Workshop on Information and Electronics Engineering *(IWIEE*), pp4053 – 4057, Elsevier, 2012, China.
[*WANG, 2013*]	H.Wang, D.Fang, N.Wang, Z.Tang, F.Chen, Y.Gu," *method to evaluate software protection based on attack modeling"* International Conference on High Performance Computing and Communications & International Conference on Embedded and Ubiquitous Computing, pp 837-844, IEEE, 2013.
[*WANGH, 2013*]	H. Wang, D. Fang, H. Dong, Y. Lei, X. Gong et Y. Gu, "Software Attack Modeling and its Application", International Conference on High Performance Computing and Communications International Conference on Embedded and Ubiquitous Computing, pp 1152-1158, IEEE, 2013.
[*ZAKR, 2011*]	A.Zakrzewska, E.Ferragut, "*Modeling Cyber Conflicts Using an Extended Petri Net Formalism*", IEEE, 2011.
[ZEIGLER, 1976]	B.P.Zeigler, "*Theory of Modeling and Simulation*", Academic Press, 1976.
[*ZEIGLER, 1984*]	B.P. Zeigler, "*Multifaceted modeling and discrete event simulation*." Academic Press, 1984.
[*ZEIGLER, 1988*]	B.P. Zeigler," *DEVS Formalism: A Framework for Hierarchical Model Development*", IEEE transactions oh' software engineering, vol. 14, no. 2, february 1988.
[ZEIGLER, 2000]	B.P. Zeigler, T.G.Kim, H.Praehofer, "*Theory of Modeling and Simulation*". Academic Press, Inc., Orlando, FL, USA, 2000.