IMT Atlantique
Dépt. Systèmes réseaux, cybersécurité et droit du numérique
2, rue de la Châtaigneraie
CS 17607
35576 Cesson-Sévigné Cedex
URL : **www.imt-atlantique.fr**

Cahier des charges

Générateur de topologies et collection de données automatisée pour FedITN

Léo Lavaur
Fabien Autrel

Date d'édition : 8 février 2023
Version : 0.0.1

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Sommaire

1. Introduction

FedITN, pour *Federated IT Networks*, s'inscrit dans le cadre de la thèse de Léo Lavaur sur l'utilisation de techniques fédérées (en particulier Federated Learning) pour l'amélioration de la détection d'intrusion dans les réseaux IT. L'objectif de ce projet est de fournir un générateur de datasets pour les réseaux IT fédérés, afin de permettre aux chercheurs de tester leurs algorithmes sur des données réalistes. En effet, la littérature actuelle sur le sujet repose sur de nombreuses facilités (données synthétiques, réseaux de taille réduite, dataset unique pour tous les participants, etc.) [1], [2], ce qui rend difficile l'évaluation de l'impact des techniques collaborative sur la détection d'intrusion.

Puisque FedITN doit pouvoir générer des datasets différents pour chaque client d'une expérimentation, il est nécessaire de disposer de topologies distinctes pour la génération des datasets. L'objectif de ce document est de spécifier les attentes et objectif du générateur de topologies, ainsi que de la partie collection de données associée.

2. Présentation du projet

L'ojectif de FedITN est de fournir un générateur de datasets pour les réseaux IT fédérés, afin de permettre des expérimentations sur des données réalistes. Pour cela, FedITN doit être capable de générer des datasets provenant de topologies de réseaux IT différentes, et de les attribuer à des clients différents. Dans cette section, nous décrivons les objectifs et le contexte dans lequel se place ce projet.

2.1. Contexte

La motivation première de ce projet est de fournir un générateur de topologies pour FedITN, ainsi que l'automatisation de la collection des données des topologies. FedITN fait partie des contributions développées dans le cadre de la thèse de Léo Lavaur sur les aspects collaboratifs de la détection d'intrusion dans les réseaux IT.

De manière plus générale, ce projet rejoint les travaux de la chaire Cyber CNI sur la conception d'un testbed holistique pour la sécurité, qui vise à fournir un environnement de test pour les chercheurs en sécurité informatique. Le testbed met notamment en avant une intégration fine entre des systèmes cyber-physiques et des SI (systèmes d'information) virtualisés, afin de tester des mécanismes de sécurité dans des environnements réalistes. Ainsi, ce testbed entièrement automatisé permettra de mener des expérimentations sur toutes les couches du modèle OSI, depuis les systèmes industriels jusqu'aux applications web.

2.2. Objectifs

Dans le cadre de FedITN, le présent projet doit cocher les objectifs suivants :

i) la génération *aléatoire* de x topologies selon un ensemble de contraintes fournies par l'utilisateur ;
ii) la simulation d'un comportement normal sur ces topologies ;
iii) l'exécution de scénarios d'attaques prédéfinis compatibles avec la topologie générée ;
iv) la collecte des données réseau (à minima au format PCAP) ; et
v) l'automatisation de l'ensemble des étapes précédentes.

2.3. Description de l'existant

Le travail sur FedITN est déjà entamé, notamment sur la définition des variations nécessaires pour construire des expérimentations robustes et pertinentes. Voici une liste non exhaustive des ressources actuellement disponibles dans le cadre de FedITN. Cette liste sera complétée au fur et à mesure de l'avancement du projet.

1. Airbus CyberRange : une plateforme de virtualisation permettant de simuler des environnements IT et OT, ainsi que de jouer des scénarios d'attaque prédéfinis.
2. 4 topologies réseau représentatives des cas d'usages considérés. L'annexe 2 est un extrait du papier en cours de rédaction sur le sujet.
3. Des pistes de travail sur la génération de topologies aléatoires, ainsi que sur la simulation de comportements normaux et d'attaques, exposées dans l'annexe 1.

3. Expression des besoins

Cette section présente les besoins fonctionnels et non fonctionnels associés au projet. Un besoin fonctionnel est une fonctionnalité que le logiciel doit réaliser, tandis qu'un besoin non fonctionnel est une contrainte sur le logiciel (performance, système d'exploitation cible, etc.).

3.1. Besoins fonctionnels

Besoin 1. *Génération de topologies* : le logiciel doit être capable de générer des topologies de réseaux IT aléatoires, selon un ensemble de contraintes fournies de manière déclarative par l'utilisateur. Par exemple, l'utilisateur doit pouvoir décrire dans un fichier de type YAML la topologie qu'il souhaite générer, en précisant :

- le nombre minimum / maximum de nœuds ;
- le nombre minimum / maximum de sous-réseaux ;
- les services minimaux (serveur web, serveur de fichiers, etc) ;
- la profondeur de l'arbre ; etc.

Besoin 2. *Simulation de comportements normaux* : le logiciel doit être capable de simuler des comportements normaux sur les topologies générées. Le comportement normal correspond à l'ensemble des flux réseau émis par les nœuds du réseau, selon un ensemble de règles prédéfinies. Par exemple, un nœud peut émettre des flux TCP vers un serveur web, ou bien des flux UDP vers un serveur DNS.

Besoin 3. *Simulation d'attaques* : le logiciel doit être capable de simuler des attaques sur les topologies générées. L'attaque repose sur l'exécution d'un scénario prédéfini qui doit être exécuté par le système. Par exemple, un scénario d'attaque peut être une attaque de type *denial of service* (DoS) qui consiste à saturer le réseau en envoyant un grand nombre de paquets vers un nœud.

Besoin 4. *Collecte des données réseau* : le logiciel doit être capable de collecter les données réseau échangées par les nœuds du réseau. Les données doivent être collectées au format PCAP à minima. Si possible, le logiciel doit être capable de convertir les données réseau au format NetFlow, en intégrant les stratégies déjà observées dans la littérature, comme CICFlowMeter [3] ou celui utilisé dans la proposition de datasets standardisés [4].

Besoin 5. *Labelisation des données* : le logiciel doit être capable de labéliser les données collectées. Le labélisage consiste à associer à chaque flux réseau une étiquette qui indique si le flux est normal ou s'il s'agit d'une attaque. Par exemple, un flux TCP vers un serveur web est considéré comme normal, alors qu'un flux TCP vers un serveur FTP est considéré comme une attaque. Puisque les timestamps des attaques sont connus, il est possible de labéliser les flux concernés. Tout ce qui n'est pas considéré comme une attaque doit être labélisé comme normal.

Besoin 6. *Documentation* : le logiciel doit être documenté de manière à ce que l'utilisateur puisse l'utiliser sans difficulté. Tout le code produit dans le cadre du projet doit être documenté.

3.2. Besoins non fonctionnels

Besoin 7. *Compatibilité* : le logiciel doit générer des topologies compatibles avec les contraintes fixées par l'utilisateur, notamment :

- l'architecture (nombre de subnets, profondeur, etc.) ;
- la liste des scénarios d'attaque requis ;
- la liste des services nécessaires.

Besoin 8. *Intégration* : le logiciel doit être capable d'être intégré dans l'environnement existant et les outils déjà utilisés. Entre autres (cette liste n'est pas exhaustive et sera amenée à être complétée) :

- la CyberRange fournie par Airbus, notamment avec LADE et les API fournies par l'outil ;
- les scenarios d'attaques fournis dans la CyberRange, bien que d'autres pourront être ajoutés ;

4. Contraintes

4.1. Délais

4.2. Contraintes techniques

5. Déroulement du projet

5.1. Planification

Cette section décrit le déroulement du projet, en détaillant les différentes étapes, classées par ordre de priorité.

a) Valider l'approche avec un set réduit de topologies. Peut-on bien déployer une sous-topologies sans configuration préalable, simplement à l'aide de protocoles réseau (DHCP, DNS, OSPF, ...) ? Est-ce que les machines peuvent bien communiquer ?

b)

5.2. Documentation

5.3. Responsabilités

Annexes

Annexe 1 – Pistes de travail préliminaires

Générer des topologies de toutes pièces est extrêmement complexe. Il est donc préférable de partir de topologies existantes, et de les composer pour obtenir une nouvelle topologie. Cela repose néanmoins sur l'hypothèse que les topologies existantes sont suffisamment variées pour permettre de générer des topologies aléatoires. De plus, les topologies en questions doivent respecter un certain nombre de contraintes, afin de pouvoir être directement composées sans reconfiguration.

Ainsi, nous pouvons définir une topologie minimale comme étant composée d'un sous-réseau contenant une machine (cliente ou serveur), et d'une gateway (ou passerelle) permettant de communiquer avec l'extérieur. Les machines définies dans la topologie disposent d'un plan d'adressage fixe. Un serveur DHCP est hébergé sur la passerelle et permet de distribuer des adresses IP aux nouvelles machines qui rejoindraient le réseau. OSPF (ou équivalent) peut être utilisé pour définir des routes entre les différentes passerelles qui seront connectées et permettre ainsi la communication entre leurs topologies.

Une topologie dite "Master" est aussi définie et sert de base pour la composition. Elle est composée d'une passerelle qui la connecte à internet, d'un point de collecte pour les logs réseau, d'un serveur DHCP pour la distribution des adresses IP, et d'un serveur DNS pour la résolution des noms de domaines à l'échelle de toutes les topologies. C'est la seule topologie qui va recevoir de la configuration, notamment pour attribuer les bons noms de domaines aux IP des machines qui hébergent des services devant être accessibles. Par exemple : serveur web (`webserver.local`), serveur de messagerie (`mailserver.local`), partage de fichiers (`fileserver.local`), etc. Cette approche permet de définir des noms de domaines uniques pour chaque service, et de les rendre accessibles depuis n'importe quelle topologie. De plus, si les scenarios impliquent de contacter une machine depuis une autre topologie, il suffit de définir un nom de domaine unique pour cette machine, et de la rendre accessible depuis la topologie Master.

Annexe 2 – FedITN (en)

2.1. Considered attacks and classification

As pointed out in introduction of this section, the dataset has to conceal two objectives: be comparable to the literature, and implement realistic attacks. Therefore, we start by comparing the most used datasets in the literature—namely NSL-KDD [5], UNSW-NB15 [6], and CIC-IDS2017 [3]—to extract relevant attack classifications. These classifications are put in perspective of the MITRE ATT&CK® [7] and D3FEND [8] matrices to provide insight on attacks, and methods for their detection. This approach ensures that the dataset is comparable with the literature, and therefore that experiments performed on it are relevant.

To provide realistic data in terms of attacks, we implement the entire attack stack in the topologies, from the attacker machine to the target, and including any necessary services, eg a DNS server for a DNS amplification attack. To that end, we rely on a cyber-range infrastructure that implement services on dedicated VMs, and complete attack scenarios through a set of executable actions.

```
1  arpspoof -i {interface} -t {target} -r {gateway} 2> /dev/null & trap 'kill \$!' INT;
   ↪ /tmp/download_swap/run.sh '{interface}' '{file}' '{regex}'; kill \$a
```

Listing 1 – Example action – ARP spoofing and file swapping

The fact that attacks are executed in a running, realistic environment ensures that the collected data is representative of real-world deployments.

2.2. Topologies

2.2.1. Topology-01

The first topology represents an advanced organization, mature in its security, and able to monitor and protect its own network. The topology is composed of five subnets (DMZ, users, administration, local servers, and monitoring) plus a WAN representing the Internet.

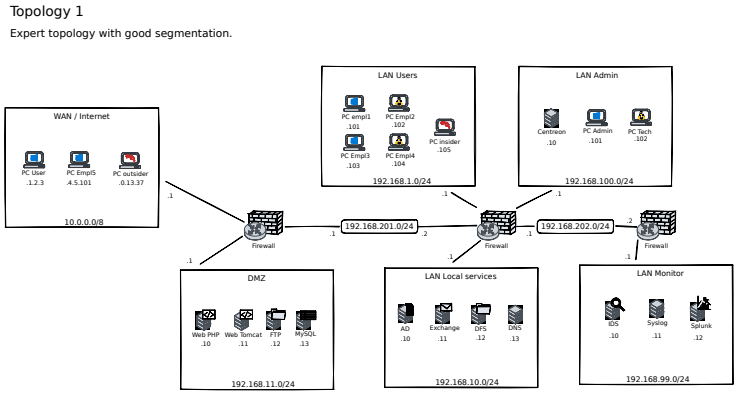

Figure 1 – Representation of `Topology-01`

Table 1 – Statistics about `Topology-01`

Property	Value
# LANs	5
# machines (hosts/server/net)	10 / 12 / 3
Monitoring	Internal (SIEM)
Attackers	2

2.2.2. Topology-02

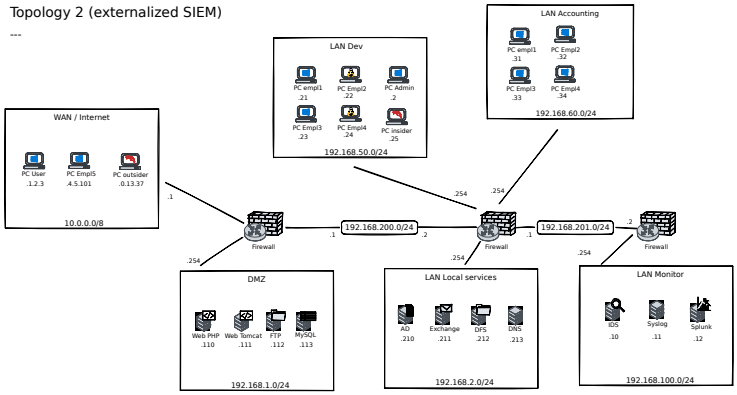

Figure 2 – Representation of `Topology-02`

Topologies 2 and 3 are similar to the first one, but differ by externalizing the monitoring. In addition to the IDS outgoing traffic that is generated, the second topology also differs by having two user LANs: one for devlopers and one for an accounting department.

Table 2 – Statistics about `Topology-02`

Property	Value
# LANs	5
# machines (hosts/server/net)	13 / 11 / 3
Monitoring	Externalized
Attackers	2

2.2.3. Topology-03

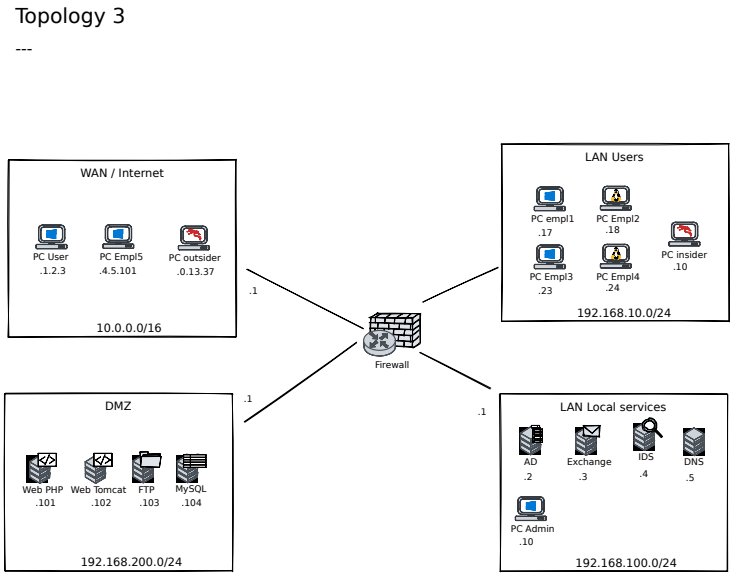

Figure 3 – Representation of `Topology-03`

The third topology is similar to the second one, but smaller and simpler. The only LANs are DMZ, users, and local servers. Monitoring is also externalized.

Table 3 – Statistics about `Topology-03`

Property	Value
# LANs	3
# machines (hosts/server/net)	9 / 8 / 1
Monitoring	Externalized
Attackers	1

2.2.4. Topology-04

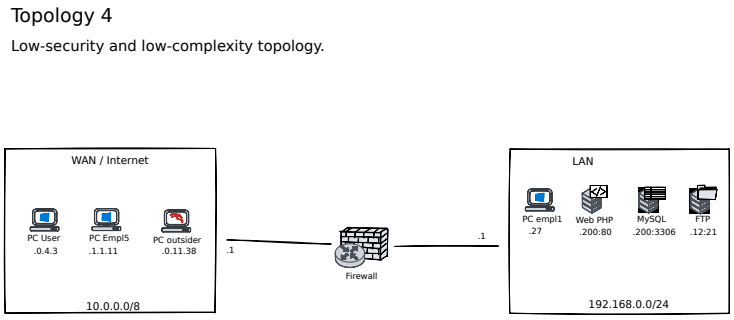

Figure 4 – Representation of `Topology-04`

The last topology represents a novice organization. Compared to the others, only few security features are in place. Only one LANs is present, for both users and exposed services.

Table 4 – Statistics about `Topology-04`

Property	Value
# LANs	1
# machines (hosts/server/net)	4 / 3 / 1
Monitoring	None
Attackers	1

Références

[1] L. Lavaur, M.-O. Pahl, Y. Busnel et F. Autrel, « The Evolution of Federated Learning-based Intrusion Detection and Mitigation : a Survey, » *IEEE Transactions on Network and Service Management*, Special Issue on Network Security Management, 2022.

[2] L. Lavaur, B. Coste, M.-O. Pahl, Y. Busnel et F. Autrel, « Federated learning as enabler for collaborative security between not fully-trusting distributed parties, » in *Proceedings of the 29th computer & electronics security application rendezvous (C&ESAR) : Ensuring trust in a decentralized world*, tex.crossref : CESAR2022.

[3] I. Sharafaldin, A. Habibi Lashkari et A. A. Ghorbani, « Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization, » en, in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, Funchal, Madeira, Portugal : SCITEPRESS - Science et Technology Publications, 2018. (visité le 14/10/2021).

[4] M. Sarhan, S. Layeghy et M. Portmann, *Towards a Standard Feature Set for Network Intrusion Detection System Datasets*, en, arXiv :2101.11315 [cs], 2021. (visité le 12/09/2022).

[5] M. Tavallaee, E. Bagheri, W. Lu et A. A. Ghorbani, « A detailed analysis of the KDD CUP 99 data set, » in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Issue : Cisda, IEEE, 2009.

[6] S. A. V. Jatti et V. J. Kishor Sontif, « UNSW-NB15 : A Comprehensive Data set for Network Intrusion Detection Systems, » *International Journal of Recent Technology and Engineering*, 2019, Publisher : IEEE.

[7] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington et C. B. Thomas, « MITRE ATT\&CK® : Design and Philosophy, » rapp. tech., 2020. (visité le 07/07/2022).

[8] P. E. Kaloroumakis et M. J. Smith, « Toward a Knowledge Graph of Cybersecurity Countermeasures, » en, 2021.

OUR WORLDWIDE PARTNERS UNIVERSITIES - DOUBLE DEGREE AGREEMENTS

3 CAMPUS, 1 SITE

IMT Atlantique Bretagne–Pays de la Loire – **http://www.imt-atlantique.fr/**

Campus de Brest
Technopôle Brest-Iroise
CS 83818
29238 Brest Cedex 3
France
T +33 (0)2 29 00 11 11
F +33 (0)2 29 00 10 00

Campus de Nantes
4, rue Alfred Kastler
CS 20722
44307 Nantes Cedex 3
France
T +33 (0)2 51 85 81 00
F +33 (0)2 99 12 70 08

Campus de Rennes
2, rue de la Châtaigneraie
CS 17607
35576 Cesson Sévigné Cedex
France
T +33 (0)2 99 12 70 00
F +33 (0)2 51 85 81 99

Site de Toulouse
10, avenue Édouard Belin
BP 44004
31028 Toulouse Cedex 04
France
T +33 (0)5 61 33 83 65

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Imprimé à IMT Atlantique
Dépôt légal : Septembre 2017
ISSN : 2556-5060