Rapport, Réorganisation du CyberRange

Pour compléter la réunion du jeudi 11 avril à 10H sur les problèmes de normalisations/organisations au sein du CyberRange, nous avons avec Allistair et Damien mis en place une solution qui nous semble fiable pour répondre aux problématiques d'aujourd'hui, qui sont :

- Le manque de visibilité dans le catalogue d'entités du CyberRange
- Le manque de visibilité des topologies dans le catalogue du CyberRange
- Le problème d'arborescence pour les entités utilisées au sein des Topologies
- Manque de suivi lors d'un « build » d'une topologie (rapport/brouillon technique)
- Manque de documentation pour les topologies en état « run »
- Collaboration et partage sur les documents utilisés dans pour le CyberRange

Proposition d'une Arborescence :

La première chose dont nous avons besoin est de mettre en place une solution d'arborescence au sein du catalogue d'entités du CyberRange pour ranger correctement les entités et les topologies de manière vraiment compréhensible pour les développeurs et les enseignants. Nous avons réfléchi à l'arborescence suivante :

Trois dossiers primaires à la racine du catalogue :

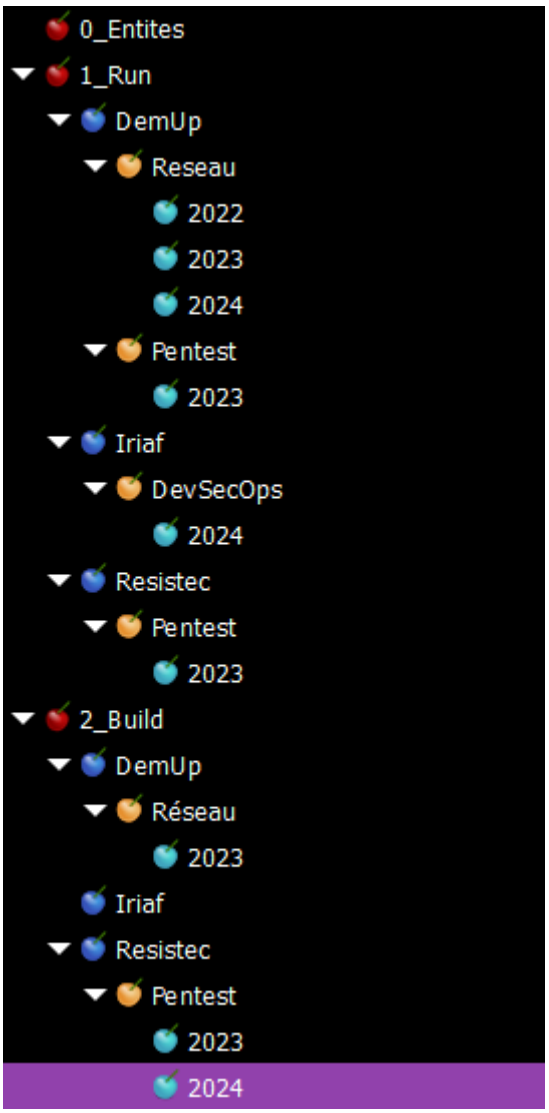
- Un dossier qui recueille l'ensemble des entités dites « stables » : Ce sont des entités qui ont été développées avec des applications et services précis dans le but d'être clonées et servir de base de référence lors de la construction d'une nouvelle topologie. (Aucune de ces entités ne doit être utilisée dans une topologie).
- Un dossier « Run » : Ce dossier regroupe les topologies utilisables pour réaliser les exercices/scénarios.
- Un dossier « build » : Ce dossier regroupe les topologies et les entités qui sont en cours de développement.

Ces trois dossiers sont chiffrés dans le but d'être rangés par ordre alphabétique au sein du catalogue :

Dans le dossier des entités, celles-ci suivront une convention de nommage dans le but d'être retrouvées plus simplement par les développeurs, nous verrons cela dans la seconde partie sur le nommage.

Dans le dossier « Run » et « Build » nous suivons la même arborescence suivante :

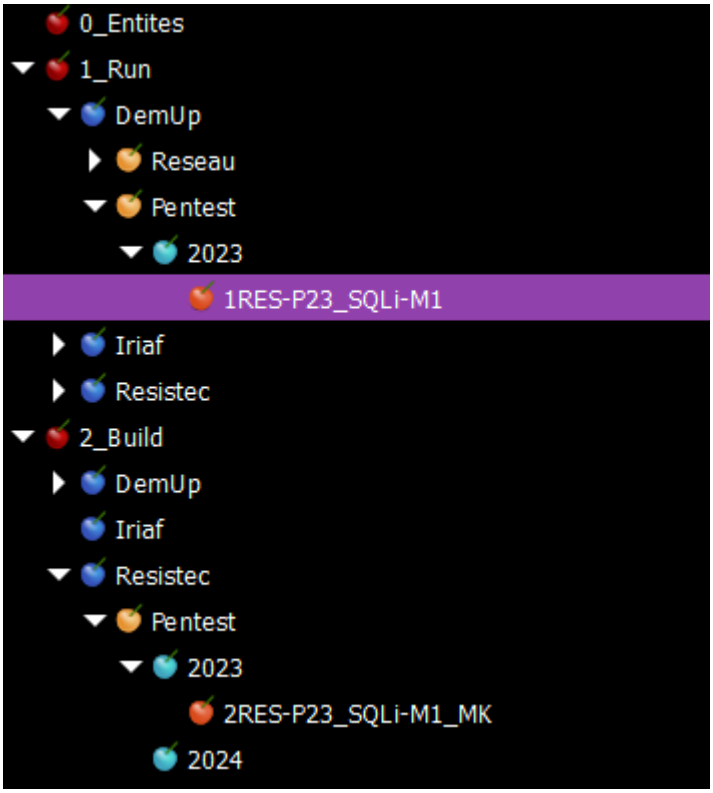
- Des sous-dossiers par service (Resistecc | DemUp | Iriaf | Diateam …)
    - Des sous-dossiers par types d'exercices (Réseau | Développement | Pentest | DevSecOps …)
        - Des sous-dossiers par années



Par la suite, on retrouve le dossier de la topologie, avec son nom normalisé qui suit le principe d'arborescence pour pourvoir ranger plus facile les topologies. On ajoute aussi à cela, le nom de l'exercice, qui est séparé par un « _ ». Le nom a pour but de décrire l'exercice (comprendre ce qui est fait à travers la topologie).

A partir de là, une déclinaison s'opère dans « Run » et « Build » car « Build » a besoin d'un peu plus de détail sur qui est ce qui travaille sur l'exercice actuellement. On rajoutera donc à la fin du nom de l'exercice un « _ » suivi des initiales (prénom, nom) du développeur de l'exercice.

Exemple :



Voici le nom de deux topologies identiques. Une est en développement et l'autre se situe dans le dossier des topologies en Run (exercice qui est fonctionnel).

Proposition d'une convention de nommage pour les topologies :

Après avoir, décidé d'une arborescence efficace, nous avons décidé de réfléchir à une convention de nommage, lié avec l'arborescence pour plus de simplicité. Nous avons deux types de nommage à définir :

- La norme de nommage des topologies
- La norme de nommage des entités

Pour la norme de nommage des topologies, nous avons décidé d'adopter la norme suivante :

[État][Service]-[Type][YY]_Nom_(Développeur)

État : Correspond à deux états pour les topologies (Run et Build). Les topologies en Build sont les topologies en cours de développement. Les topologies en « Run », sont les topologies qui ont été vérifiées et testées, elles sont considérées comme jouables. L'état est attribué via un chiffre soit 1 (pour « Run », voir arborescence), soit 2 (pour « Build », voir arborescence), cela correspond à l'ordre alphabétique dans l'arborescence.

Service : Correspond aux différents services qui utilisent le CyberRange. Attribué sur 3 lettres on retrouve :

Service	Acronyme (1 lettres)
Sciences des risques et de la donnée (si appartient à plusieurs services)	SRD
Resistecc	RES
DemUp	DUP
Diateam	DIA

Type : Correspond au type d'exercice principal de la topologie. Par exemple nous avons des topologies plus orientées sur des exercices de Réseau, de Pentest, de DevSecOps… Une lettre est donc utilisée pour décrire le type d'exercice.

Type d'exercice	Lettre associée
Réseau	R
Programmation	P
Pentest	H
DevSecOps	X
Forensic	F

YY : On ajoute à la suite les deux chiffres indiquant l'année ou l'exercice a été développé. Cela permettra d'affiner les recherches sur les exercices développé au fur et à mesure des années.

Précédé d'un « _ » on ajoute le nom de la topologie, le nom a pour but d'expliquer ce qui est fait durant l'exercice.

Enfin, on fonction de l'état de l'exercice, si celui-ci est en état de développement, on ajoute à la fin du nommage les initiales du développeur travaillant sur la topologie. Cela permet de tout de suite savoir, si le « build » en cours est toujours d'actualité. (Différenciation entre les stagiaires et les administrateurs …).

En cas de modifications les topologies en « Run » doivent-être clonés dans le répertoire « Build » adapté avant qu'il y ait des modifications apportées.

Mettre dans le nommage l'année et plus globalement dans l'arborescence, permet lors de la recherche d'une topologie de faire rapidement la distinction. Pour les topologies en « build », les administrateurs à la fin d'une année peuvent faire un tri pour supprimer les topologies qui n'ont pas abouti, ou faire évoluer d'une année à l'autre les topologies qui sont encore en développement.

Voici quelques exemples :

Une topologie en cours de construction par Hamza Romdhani sur l'exploitation et la réparation des mauvaises habitudes de codes en 2024. L'exercice est à destination des étudiants, donc il appartient au service DemUp. On retrouve donc le nommage suivant :

2DEM-H24_ExploitRepairWebCodes-SFA_HR

Si l'exercice est fini d'être développé et est ajouté au dossier des topologies « Run », alors on aura le nommage suivant :

1DEM-H24_ExploitRepairWebCodes-SFA

Une topologie qui est fini, utilisable et destinée aux étudiants pour apprendre le principe de routage en réseau. Développée en 2023 Par Mathieu Koltok dans le cadre de DemUp :

1DEM-R23_Routing-M1MRSI

Pour le nommage des topologies celui-ci doit d'abord être apporté au sein du catalogue des topologies. La convention de nommage permet de retrouver facilement les topologies en « Build » ou « Run » en fonction (du type et du service) de l'année, de son nom et enfin par développeur pour les topologies en « Build ». Le nommage est important, car il n'y a pas d'arborescence dans le catalogue de topologies. Grâce à la barre de recherche dans le catalogue, il est très rapide de filtrer les topologies en fonction d'un ou de plusieurs de ces paramètres.

Un dossier portant le bon nommage doit-être créé dans le répertoire adéquate, les entités seront ajoutées par la suite dans ce répertoire.

Lors de la suppression d'une topologie le procédé inverse doit être respecté. On peut supprimer la topologie avec l'ensemble des entités automatiquement, mais il ne faut pas oublier de supprimer le dossier portant le même nom que la topologie venant d'être supprimée.

Le nommage est très utile pour la création de script d'automatisation. On pourrait imaginer une interface nous demandant de choisir un nom pour la topologie, puis on sélectionne le type, le service, le développeur. La topologie et le dossier dans le catalogue d'entité serait donc créée avec le bon nommage. De la même façon on pourrait imaginer le script de suppression, en sélectionnant via l'interface la topologie à supprimer.

Proposition d'une convention de nommage pour les entités :

Pour le nommage des entités, on retrouve en premier l'état, ici soit « Build » (2) ou alors « Stable » (0). Une entité est stable ou en « build », elle ne peut pas être en « Run » car la convention de nommage ne s'applique par aux entités en « Run ». Les étudiants, les utilisateurs de l'exercice doivent comprendre en un mot l'utilité de l'entité, pour cela on ne peut pas avoir de convention de nommage en « Run ». Cependant dans la description de cette entité on retrouve le nommage de l'entité dans son état « Stable ». Si l'entité en « Run » doit repasser en phase « Rebuild », alors on doit trouver rapidement dans sa description son origine.

Après son état, on retrouve type, « S » pour serveur, « C » pour client et « I » pour équipement intermédiaire, suivi de son OS et de la version de celui-ci (voir si obsolète). Par la suite, on ajoute son nom qui permet de comprendre son utilité, précédé d'un « _ ». Enfin, on peut ajouter les initiales du développeur si l'entité est en « build » et a pour but d'être ajouté dans le dossier des entités stables.

Dans la description d'une entité on retrouve les services et les applications installés sur celle-ci et pouvant-être utile dans la construction d'une topologie.

[État]-[OS][Version]_Nom_(Développeur)

État : Soit « Stable », correspondant à « 0 », soit « Build », correspondant à « 2 ».

OS : Les différents Systèmes d'exploitation pouvant-être utilisés dans le CyberRange. Cela permet de comprendre sur quoi nous allons travailler en un coup d'œil. De plus souvent les équipements intermédiaires open-source sont basés sur des distribution Linux.

Version : La version de l'OS qui est utilisé. De cette manière on peut faire évoluer les exercices au fur et à mesure des années en mettant à jour les différentes versions d'OS. On peut aussi se rendre compte de l'obsolescence des OS.

Exemple :

Un Serveur Ubuntu version 24.04 dans le dossier des entités stables avec les services Apache MySQL et PHP.

0Ubuntu24.04_LAMP

Un routeur VyOS version 1.1.8 utilisé dans une topologie en « build » par le développeur Mathieu K.

2VyOS1.1.8_Routeur_MK

Un client Windows 11 utilisé dans une topologie en « Run » :

Windows11

Dans les topologies en « Run » les entités sont nommées explicitement pour être reconnaissable facilement par les joueurs. Cependant on garde dans la description le nommage de la topologie durant son état « Stable ». La description sert donc a retrouvé facilement la machine précise, sa version qui est utilisée. De plus dans la documentation (la prise de notes lors de la construction de la topologie « build ») on peut facilement faire le lien avec l'entité de départ pour retrouver les modifications apportées.

De la même manière qu'avec le nommage des topologies, celle-ci sert à trier les éléments au sein du CyberRange les entités peuvent-être repérer plus facilement par les développeurs tout en connaissant leur origine. Si les entités sont bien nommées, en cas d'erreurs on peut utiliser la recherche par filtre pour retrouver efficacement les entités. Cela permet aussi de faire facilement de la place sur le CyberRange et de ne pas supprimer par erreurs des entités stables ou dans les topologies « run ». Limiter le nombre d'entités dans le CyberRange permet une meilleure optimisation des performances donc il ne faut pas négliger le nettoyage fait dessus.

Une topologie nouvelle, est construite à partir d'entités provenant du dossier des entités stables. Pour cela il suffit de cloner les entités depuis le dossier des entités stables dans le dossier de destination « Build ». Lors du clonage on peut donc choisir le bon répertoire au clonage et aussi changer son nom directement à ce moment-là. De cette manière le développeur choisi les entités qui lui sont nécessaires et les renomme selon la convention « build ».

Si un service ou une application doit-être ajouté, on regarde dans un premier temps dans quelle entité il peut être ajouté avant de créer une nouvelle entité. La création d'une nouvelle entité provoque la création d'un nouveau document lié à cette entité ainsi que la référenciation de celle-ci dans le logiciel de documentation. On cherche donc a ajouté le service/application dans un premier temps dans une entité existante, ainsi on complète que le document de référence de cette entité. Pour ajouter un service on suit le protocole de maintien des entités qui consiste à exporter l'entité du CyberRange pour travailler sur la machine virtuelle depuis l'extérieur.

Passage de « Build » à Run et Run à « Build » :

Pour commencer un passage d'une topologie « Build » à « Run » permet de définir une topologie qui était en état de construction à l'état stable, l'état utilisable en cas d'exercice. Cela comprend un accord commun entre les administrateurs sur sa réalisation. Après des phases de tests sur la topologie, les entités peuvent-être renommées dans leur nom compréhensible pour un joueur, tout en conservant en description la convention de nommage pour un « rebuild » facile.

Par la suite un dossier venant accueillir les entités de la topologie doit-être créé dans le répertoire « Run » correspondant. Par la suite on clone cette topologie, en la renommant avec son futur nom de « Run ». On ne peut pas choisir ou le clonage va apparaitre dans l'arborescence. Le clonage apparait obligatoirement à la racine de l'arborescence. On peut donc repérer facilement le nouveau dossier qui est apparu à la racine avant de déplacer son contenu dans le dossier créer dans le « Run » dédié à la nouvelle topologie « Run ». Par la suite il nous reste à supprimer le dossier de la topologie présent à la racine puis supprimer la topologie et les entités en « build » depuis le catalogue de topologies. Pour finir on supprime le dossier « build » qui accueillait les entités de la topologie en « Build ».

Pour le passage de Run à Build, on effectue exactement les mêmes étapes mais vers la destination d'un répertoire « build ». Pour le nommage des entités, suffit de reprendre le nom présent dans la description de l'entité et de rajouter les initiales du développeur. Pour des questions de sécurité, cette fois-ci on ne supprime pas la topologie en « Run », on la remplacera seulement après avoir effectué les modifications dans le « Build » en suivant les étapes de passage de « Build » à « Run ».

Clonage d'une topologie :

Pour le clonage des topologies dans le cadre d'un exercice, il suffit de cloner directement une topologie présente dans le catalogue des topologies autant de fois que l'on veut et de laisser le nom avec « _cloneX » à la fin. Tous les clones apparaissent à la racine ce qui nous arrange dans ce cas. Une fois les exercices finis on peut faire une recherche dans le catalogue avec le filtre : « clone » pour que tous les clones apparaissent, on peut donc les supprimer facilement. Cela fonctionne que si les règles précédentes ont été correctement respectées.

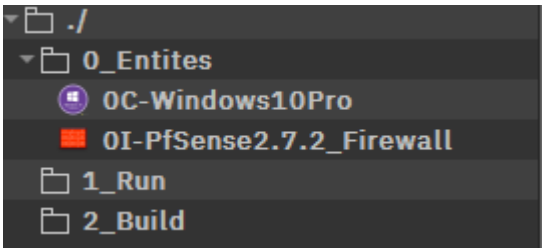
Par la suite, dans le dossier racine on peut supprimer tous les dossiers qui contenaient les topologies clonées dans le cadre d'exercices.

Colorisation des topologies :

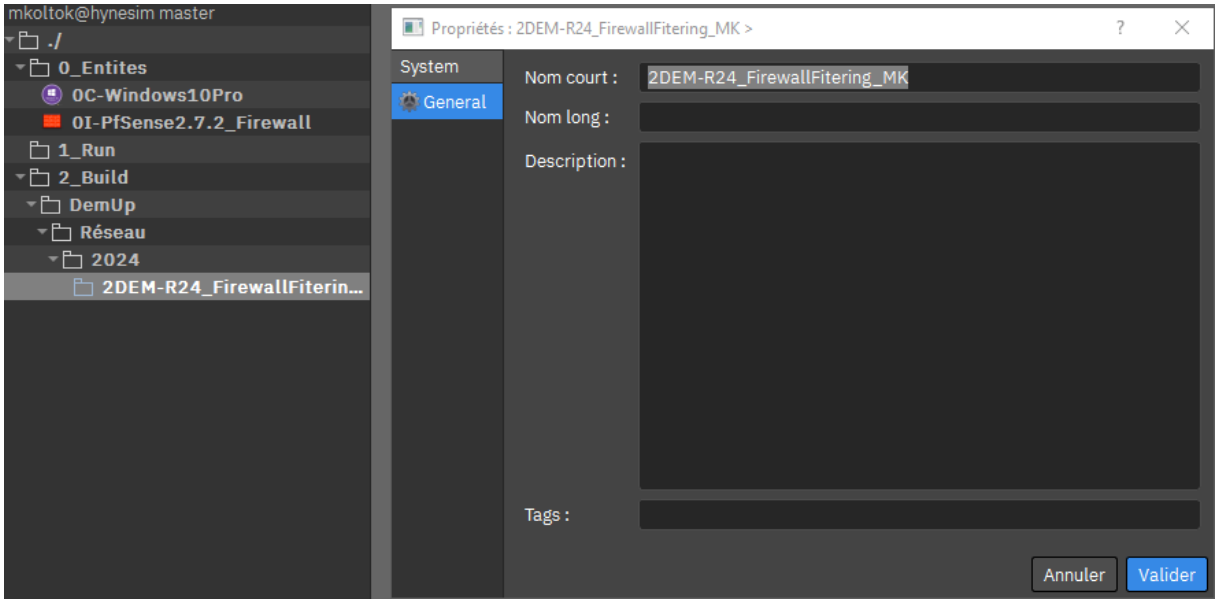
Pour discerner encore mieux les topologies dans le catalogue des topologies, nous avons imaginé un code couleur de fond des topologies. Cela serait surtout utile pour les administrateurs, pour différencier les topologies dans le catalogue des topologies facilement. Les topologies en « Run » auraient une couleur appropriée, tandis que les topologies en « Build » auraient des couleurs différentes en fonction du service qui est en cours de développement.

Exemple de toutes les conventions :

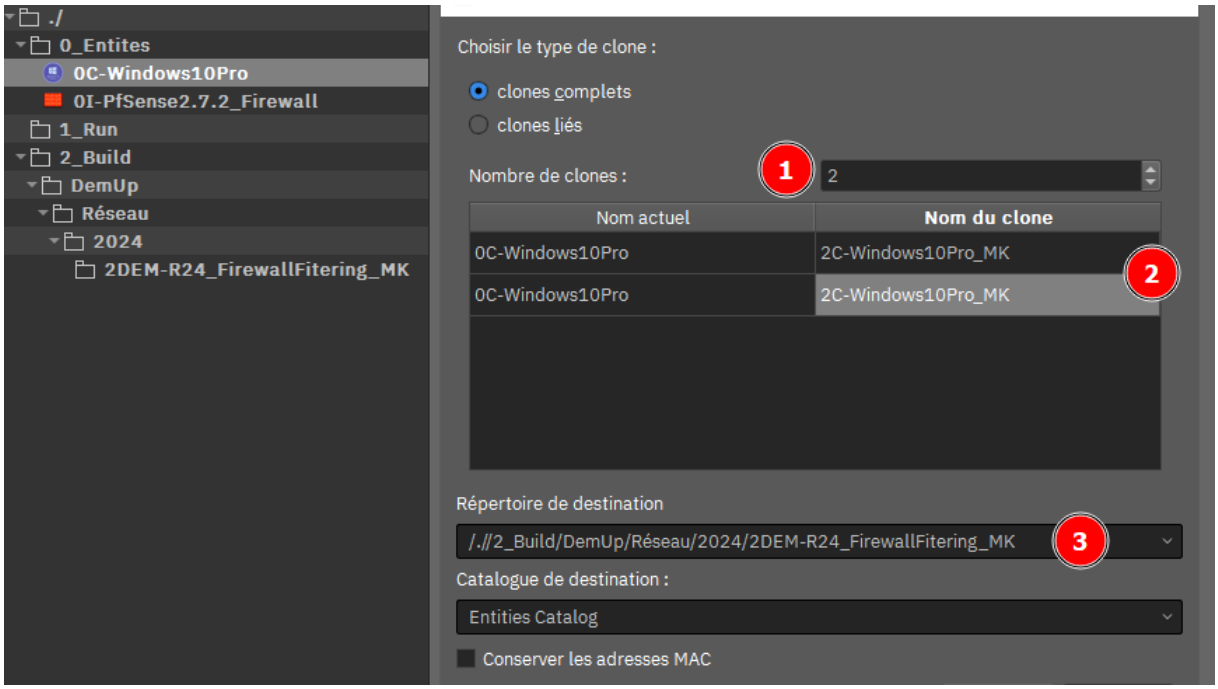
Voici la racine avec pour le moment deux entités stables (un pare-feu, un client windows 10). On va créer une topologie pour un scénario de réseau à destination de DemUp.



On commence par créer les répertoires dans le « build » venant accueillir la nouvelle topologie.

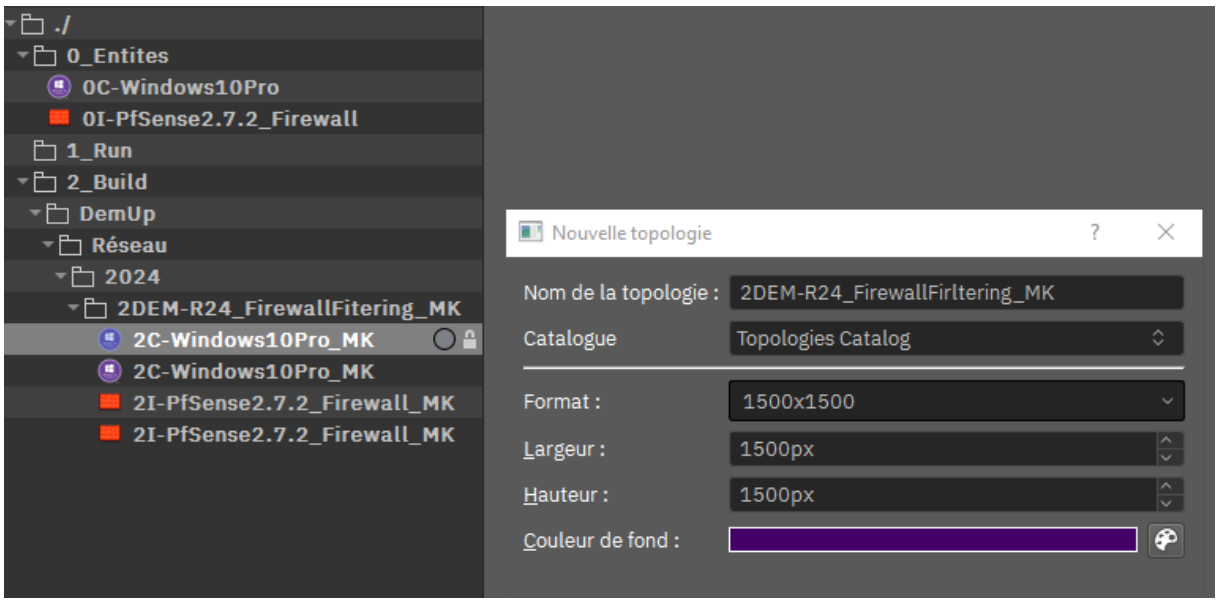


Une fois que le dossier venant accueillir la topologie est créé, on va cloner les entités pour réaliser la topologie depuis le dossier des entités stables. Imaginons que j'ai besoin de 2 clients Windows 10 et deux Firewall :

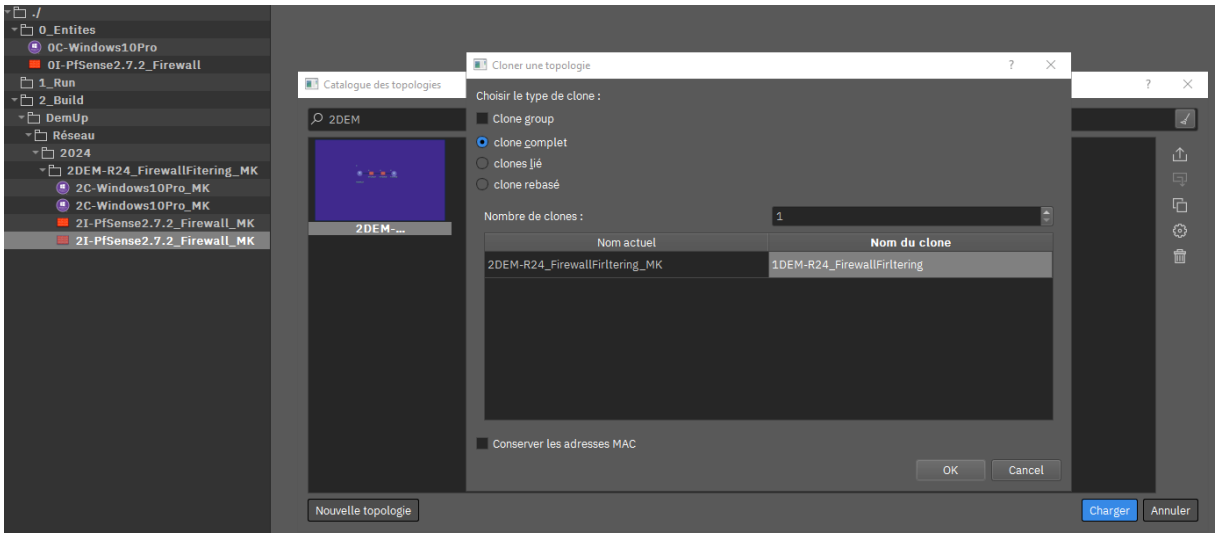


En « 1 », on choisit le nombre de clone. En « 2 » on les renomme pour qu'ils aient un nom associé à une topologie correspondant à « build ». En « 3 », on choisit le dossier correspondant à la topologie build pour accueillir les entités. On fait la même chose avec le firewall.

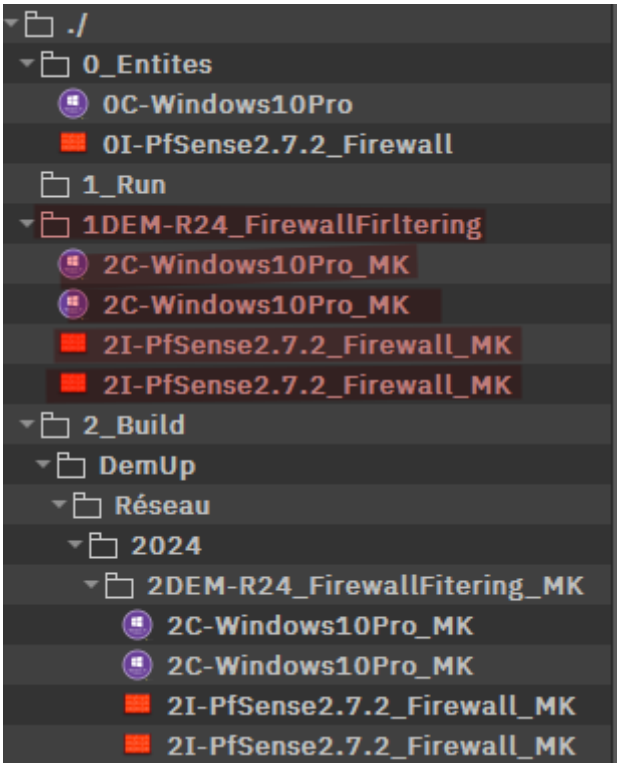
Par la suite on peut construire la topologie et travailler sur le scénario. Pour cela on va créer une nouvelle topologie avec le nom et la couleur associée. :



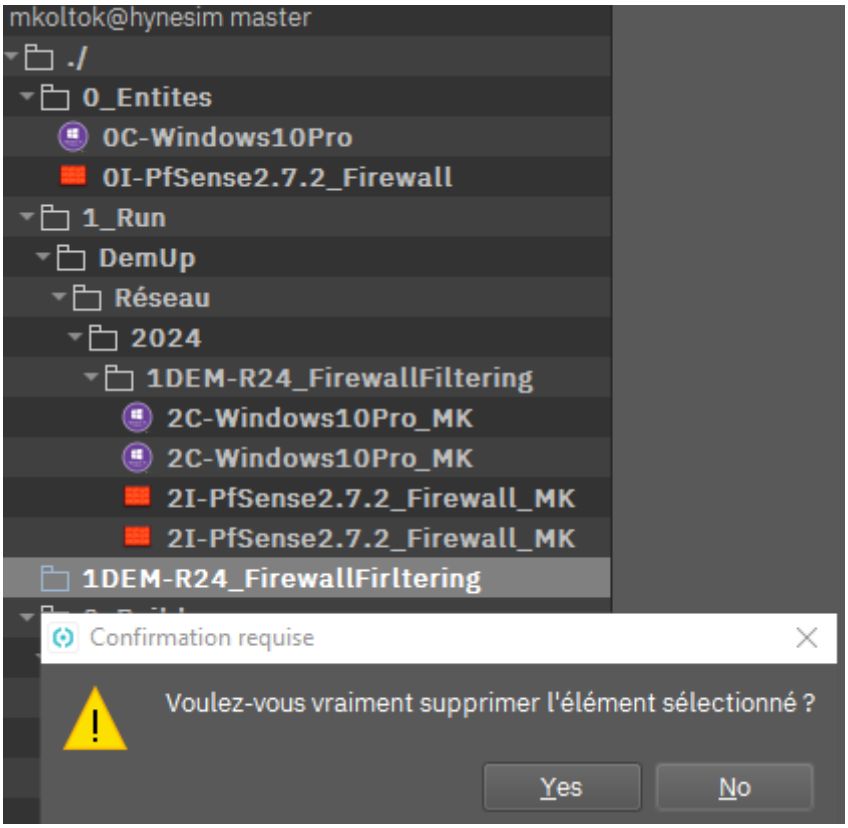
Une fois la topologie terminée, on va la cloner pour la déplacer en état « Run ». Pour cela on peut retrouver facilement la topologie dans le catalogue grâce à son nom, la décharger et créer un clone avec son futur nom de topologie « Run ».



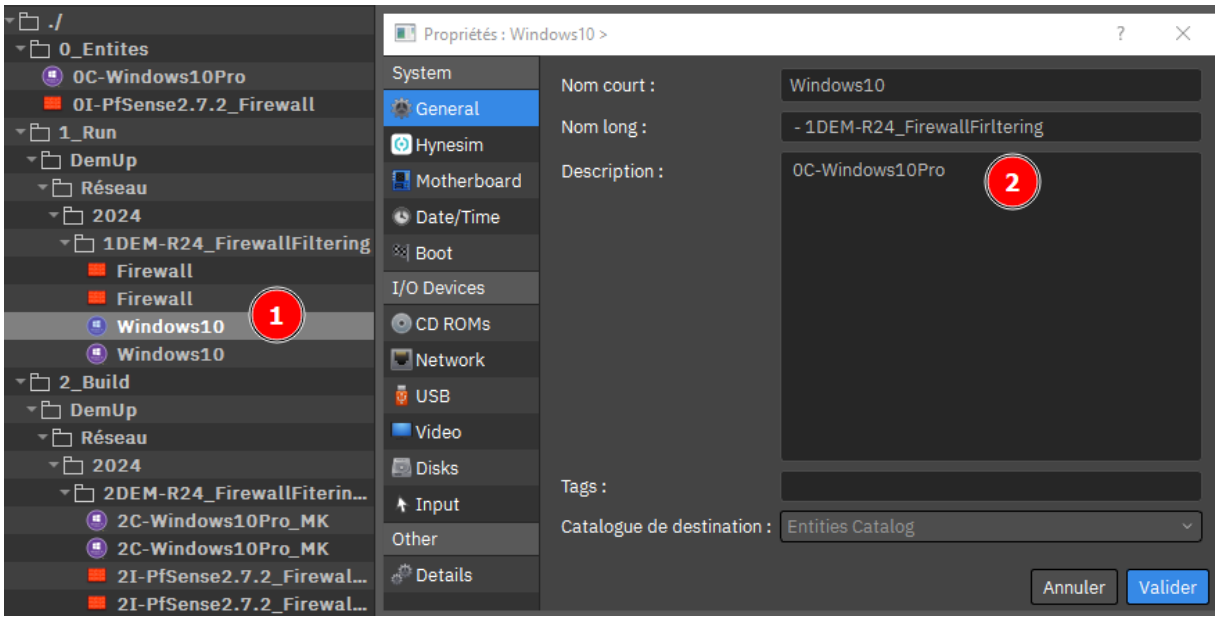
Lorsqu'on effectue le clone, celui-ci apparait au niveau de la racine, car on ne peut pas choisir le répertoire de destination du clone de la topologie.



On voit la topologie en rouge, donc il faut créer le répertoire dans « build » pour venir mettre la topologie au bon endroit de l'arborescence. Puis on déplace les entités avant de supprimer le dossier présent à la racine.



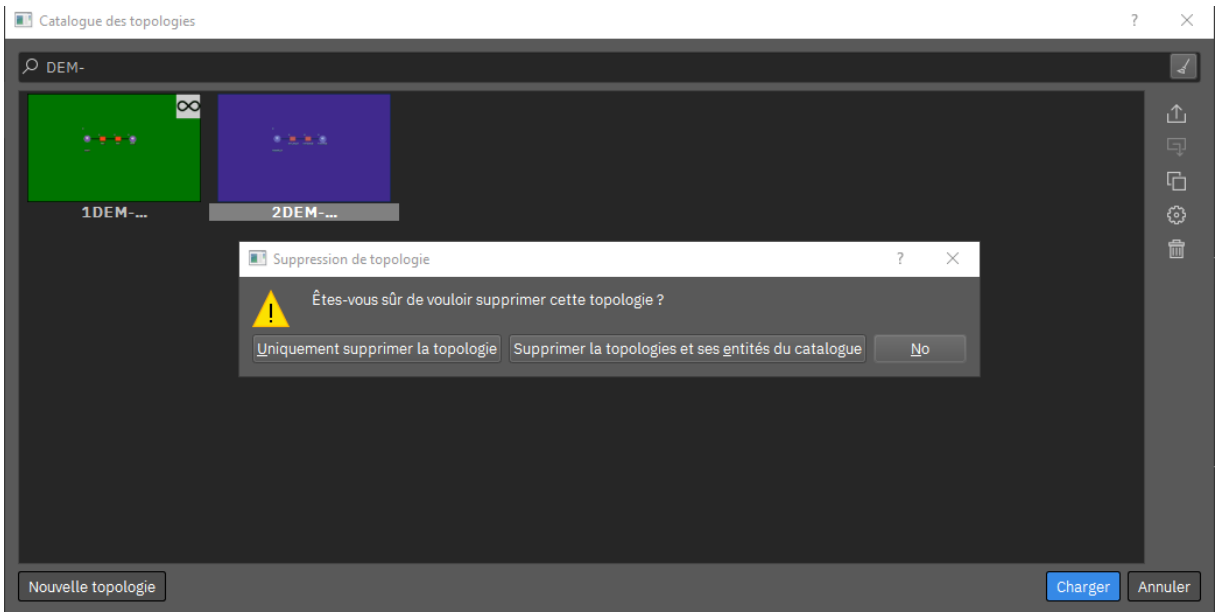
Nous devons maintenant trouver un nom adapté aux entités pour que les joueurs comprennent directement quel est le rôle de celle-ci.



En « 1 », on retrouve le nom des entités pour les joueurs et en « 2 » on peut retrouver le nom de l'entité dans son état « stable »

Il suffit maintenant de changer la couleur de fond pour la faire correspondre à l'état d'une topologie en « Run ». Voici, elle est clonable, jouable ….

Il ne reste plus qu'à supprimer la topologie dans l'état « build » qui n'est plus utile, pour ne pas faire d'erreur, il faut retrouver la topologie dans le catalogue des topologies puis supprimer les entités associées :



On peut donc voir que la topologie en « build » n'existe plus, il faut enfin supprimer le dossier présent dans « build » qui est maintenant vide.

Pour faire un passage de Run à Build, on effectue les mêmes étapes. Mais pour des questions de sécurité il vaut mieux ne pas supprimer la topologie en « Run » avant d'avoir effectué les changements dans la topologie.