Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage et qui m'ont aidé lors de la rédaction de ce rapport.  
  
Tout d'abord, j'adresse mes remerciements à mon professeur, **Mr Chrsitian TOINARD,** tuteur de stage.  
  
Je tiens à remercier vivement mon maitre de stage, **Mr Philippe CHARRON,** pour son accueil, le temps passé ensemble et le partage de son expertise au quotidien. Grâce aussi à sa confiance j'ai pu m'accomplir totalement dans mes missions. Il fut d'une aide précieuse dans les moments les plus délicats.  
  
Je remercie également toute l'équipe de Telespazio Guyane pour leur accueil, leur esprit d'équipe et en particulier **Mr Christophe VIALON et Mr Julien …**, qui m'a beaucoup aidé à comprendre les problématiques d'achats sécurisés...  
  
Enfin, je tiens à remercier toutes les personnes qui m'ont conseillé et relu lors de la rédaction de ce rapport de stage : ma famille et un collège de l’INSA Rennes Mr Jérôme …

Résumé

Le but était que le stagiaire étudie la possibilité de migrer les règles existantes d’un des firewalls principal (type **ciscoasa**) du **CSG (Centre Spatial Guyanais**) (que l’on appellera **PRIN**) dans le logiciel **google/capirca** et de mettre en place les mécanismes adéquats permettant la gestion de configuration des demandes flux utilisateurs.

Cette demande s’articulant autour de 2 axes principaux. La mise en place d’un mécanisme de génération des règles et leur gestion en configuration (git ou subversion ?) à partir des demandes de flux utilisateurs. Mais aussi l’adaptation des règles existante pour les injecter dans le système **google/capirca.**

Capirca développé par Google, d’où **google/capirca** permet la génération d’acccès listes de plusieurs types (cisco, junipers, iptables …) et bien entendu **ciscoasa**, ce qui nous intéresse ici.

Au travers de différentes spécifications techniques il a était proposé pour répondre à cette expression du besoin la mise en place d’un serveur git open source, **gogs** (go git service), embarquant **google/capirca** et d’une interface utilisateur de type shell/python pour la gestion des demandes de flux utilisateurs. Interface prenant en compte la gestion de tous conflits qui pourrait exister entre le manager (**cisco asdm**), et les différents collaborateurs. et En effet, les access lists étant des données ultrasensibles, plus encore lorsque l’on parle d’activités spatiale, il est totalement interdit d’utiliser un serveur tel que gitHub ou gitLab.

Pour des questions évidentes de sécurités et de précautions il a était mise en place une maquette pour la réalisation et la démonstration de ce projet. Cette maquette ressemblant en tout point à la réalité à la différence près des adresses ips.

Le résultat final tel que remis permet à tout gestionnaire présent dans le réseau du **PRIN** de cloner le projet contenant l’interface de gestion et toutes les nouvelles demande de flux d’en ajouter ou de modifier les existantes. Toujours à l’aide de l’interface la nouvelle configuration peut ensuite être envoyée dans la running-config. Deux gestionnaire peuvent travailler sur les deux modes d’intégration existants (**google/capirca**, le **manager**), sans s’inquiéter des conflits pouvant être créé. Cette solution permet donc une réponse plus rapide mais aussi une gestion plus organisé des demandes de flux utilisateurs.

Glossaire

* <https://github.com/google/capirca/wiki>
* <https://gogs.io/docs>
* <http://www.cnes-csg.fr/web/CNES-CSG-fr/>
* <https://www.telespazio.fr/fr>
* <https://pynet.twb-tech.com/blog/automation/netmiko.html>