Bonjour Aurélien, peux-tu te présenter ?

Je peux commencer par mon parcours. Un touche à tout autodidacte. J'ai débuté par la plus petite porte comme technicien dans un call center d'HP (l'approche social engineering comme je l'aime à l'appeler), et puis j'ai fait mon petit bonhomme de chemin, tout d'abord dans les systèmes, puis dans le domaine de la sécurité en continuant en parallèle les cours du soir au CNAM. Mon meilleur souvenir ? Le développement d'un SIEM LDAP from scratch pour EDF. Le premier du genre en 2008 fonctionnant sous RedHat, Aix et Sun. Toujours aussi pratique, et surtout opensource ;) Qui aurait pu savoir que les docs sur lesquelles je m'appuyais 5 ans plus tôt provenaient de celui qui deviendrait mon collègue 5 ans plus tard. Chaque expérience, même si repoussante aux premiers abords, apporte un plus. Ce qui est frappant lorsque l'on commence tout en bas de l'échelle, ce sont les mauvaises pratiques que l'on voit chez ceux qui sont au-dessus. Lorsque l'on est parachuté ingénieur, sans passer par les étapes d'avant, on ne peut pas comprendre les personnes qui nous entourent et leurs pratiques. Je me suis nourris de ces expériences pour arriver là où j'en suis aujourd'hui. Toujours se remettre en question. Ne pas vivre sur ces acquis. Aujourd'hui RSSI, je me rends compte que le fruit de mon expérience provient des mauvaises pratiques des uns et des autres.

- Tu as mis en place une plateforme BigBlueButton (BBB pour les intimes) pour permettre aux enseignants de l'école de donner leurs cours à distance. Quels sont selon toi les avantages de cette solution (possiblement par rapport à d'autres plateformes) ?

BBB est une solution opensource développée au Canada par la société Blindside Networks, des développeurs Kurento, et développeurs coturn.

En préambule, voilà comment tout commence. J'apprends l'existence de BBB pour la première fois en réunion de service. Il était question de mémoire de trouver des outils se rattachant à Moodle. Tout de suite mon esprit de curiosité me pousse à installer une instance sur une machine de DEV. A ce moment là, voyant la version d'OS demandée (Ubuntu 16.04 LTS), le RSSI que je suis se demande si le produit est bien "secure"; je comprends par la suite par l'historique de BBB, que la sous-couche flash que les Devs trainent les contraints à suivre dans cette distro en attendant la prochaine release majeure. Mes premiers pas n'ont pas été de tout repos. toutes les technos employées (freeswitch / nginx / nodejs / kurento / redis / coturn / webrtc) étaient du domaine de l'inconnu pour moi. On peut vite comprendre que BBB n'allait pas être simple à prendre en main https://docs.bigbluebutton.org/2.2/architecture.html.
Mais bon, si cet outil devait voir le jour, il fallait creuser. Très vite on se rend compte que BBB est une coquille vide sans un frontend. Le LTI Moodle devait normalement faire le boulot, mais dans le cadre de mes tests, je n'allais pas monter un Moodle. Alors c'était parti pour l'applet java de demo... Je ne vais vous mentir, je n'aime pas java (et au passage encore moins python plus ou moins pour les même raisons de "portabilité" théoriques, bref, passons). Bon la demo c'est bien gentil, mais peut-on aller plus loin ? Qui a t-il sur le marché comme frontend ? Et bien pas grand chose... Tout de suite on se met à chercher. Les contraintes ? Un outil qui accepte à minima un SSO ou encore mieux une fédération d'identité. Je finis par identifié deux choses : mconf et greenlight. mconf est "partenaire" de BBB pour la redistribution du code, je m'attendais donc à trouver chaussure à mon pied. On met en place l'instance, on teste, et on se rend très vite compte que l'outil est très vieux et plus adapté. Je me tourne donc vers greenlight, livré avec une couche docker... Ayant horreur de docker (je ne m'en cache pas), je décide d'alpaguer les devs de BBB pour savoir comment "sortir" la solution de la boite. Première difficulté : Ruby ! Encore une découverte (On est en Mars 2020...)
Allez, je me lance, je me fais une "formation" maison (autodidacte), et j'arrive très vite à comprendre comment ROR fonctionne (je le conseille pour les Devs qui débutent). Ayant levé toutes les difficultés, j'avais un produit en DEV/préprod qui commençait à ressembler à quelque chose. Le RSSI que je suis pense tout de suite à l'adhérence d'un tel outil avec le premier confinement qui est instauré (vs Zoom, Skype, Teams, Discord...). Je joue très vite la carte d'un service proposé non pas à l'échelle du Campus, mais à celui de l'IMT. Ayant déjà prévu le coup avec l'auth par la fédération, il fallait maintenant rendre l'outil corporate. Par chance, tous les supports graphiques sont à dispo. J'enfile donc ma casquette de webdisigner pour customiser greenlight et BBB. Je finis par proposer la solution.
A refaire, je mettrais d'abord des métriques avant de la "vendre"/proposer (on apprends de ses erreurs). Je me rapproche très vite du DPO de l'IMT pour valider tous les aspects RGPD (sur sol français, dans nos murs, accès restreints, protection des données personnelles, consentement, modération des contenus, ...). La solution prend

presque tout de suite lors des premiers amphi du personnel. La question du dimensionnement est très important. On fait un premier amphi à 230 personnes, et ça tient !

Les avantages en quelques mots : Facilité de mise en œuvre d'une solution opensource (malgré le nombre non négligeable de techno derrière). La documentation, pour peut qu'on se donne la peine de lire et de chercher un peu, est très bien faite. J'invite fortement les personnes s'intéressant à BBB à s'abonner aux issues github, qui regorgent d'informations. La solution peut être totalement cloisonnée dans un réseau privé. Je crois qu'il n'y a pas beaucoup de plateforme du genre qui s'interface avec un trunk SIP pour proposer un pont audio. On parle beaucoup de la plateforme, mais ne jamais oublier dans la boucle les développeurs. J'ai pu apprécier la qualité des échanges techniques sans payer de support. Jitsi-meet était un candidat aussi, mais excusez moi, voir lorsque je fais une visio à 25, si je ne peux voir que 4 personnes (à l'image de Teams en Mars 2020), bah non, je suis frustré. Et puis je ne parle pas de la qualité de l'image. On croyait voir un burst de flux plein de pixel, à l'image des premières vidéos encodées dans les années 2000... jitsi-meet est sans doute sympa lorsque l'on a recodé tout l'outil...
Bon maintenant, pourquoi pas youtube ? Discord ? Zoom ? Teams ?
Pour contrer Youtube, un flux live rtmp (BBB live) a très vite vu le jour. Donc pas besoin d'aller voir ailleurs.
Pour discord ? Est-ce vraiment fait pour enseigner ? (ça se passe donc de commentaires)
Pour Zoom ? (malgré l'achat massif de licence par certaines entités, on ne m'enlèvera pas de la tête tous les contenus inappropriés qui ont circulé pendant des réunions assez sensibles, le manque de sécurité de l'outil, le partage des clés de chiffrement avec la Chine, le passage obligé des flux par les US... et j'en passe).
Pour Teams ? Pas de bras, pas de chocolat; Pas de client, pas de "rendu" correcte. Le client étant un mouchard en puissance, vous connaissez mon point de vue. Merci encore à celui ou celle à l'échelle de L'État qui a donné son accord pour que Microsoft investisse les lieux de l'Éducation Nationale...
Pour faire simple, étant RSSI, il ne restait pas beaucoup de choix; une solution payante certifiée ANSSI (à quel prix ??), ou bien BBB en interne. Lorsque la crise est arrivée, je dirais comme de manière générale, le budget alloué à la SSI est quasi nulle. Donc je vous laisse conclure par vous même des choix qui me restaient :D Je pense que cela n'est pas courant d'avoir un RSSI, qui gère les systèmes, les réseaux, la téléphonie, le développement, et j'en passe. Ayant toutes ces casquettes, il était beaucoup plus facile pour moi d'être proactif.

- Du point de vue de la sécurité, et en prenant ta casquette de RSSI, quelles mesures as-tu mises en oeuvre pour améliorer la sécurité de la plateforme BBB ?

J'ai très vite compris qu'il fallait s'abonner à ce qui touchait à BBB sur github. Un produit qui émerge aussi vite est successible d'attirer des mauvais esprits. je tombe très rapidement sur des binaires permettant de faire tomber BBB... Je fais donc en sorte de me protéger en sensibilisant les utilisateurs de la plateforme (du moins aussi bien que je pouvais), de protéger si possible les salons, d'éviter de laisser traîner les urls des salons sur Internet... La sécurité c'est 50% de social et 50% de technique. Encore une fois, on ne peut pas tout bloquer.
Très vite j'ai été victime de mes "bonnes pratiques" : On ferme tout et on ouvre ce dont on a besoin... Une plage très large de ports UDP, et surtout la couche IPv6 qui n'arrangeait pas mes affaires... Après quelques erreurs 1007 (private joke), et quelques salons sans micro, j'ai réussi à maîtriser les flux entrants en sortants.
Je me suis aussi très vite rendu compte que les devs de BBB ne mettaient pas à jour tous leurs outils (versions obsolètes de Ruby..., trous béants dans des confs nginx...). Le RSSI que j'étais a très vite corrigé cela; ayant fait mon auto-formation sur les outils, je savais que les mettre à jour ne risquait pas de casser leur fonctionnement, donc autant mettre à jour. J'ai aussi désactivé tout ce qui ne servait à rien. Greenlight arrive avec tout un tas de moyen de s'authentifier, ce qui alourdit de mon point de vue le spectre des problèmes de sécu. Autant les supprimer tout simplement si ça ne sert pas. Bon il faut manger un peu de Ruby... Confinement = piscine ? (private joke). J'en ai profité pour publier mon code pour l'intégration de Shibboleth dans Greenlight.
Supprimer docker ! Comme une mission que je m'étais donné. Tout est livré dans l'infra BBB à base docker par ci, docker par là. Et bien non, pas besoin de la couche Docker. Cela n'a pas été chose simple partant de zéro. Mais avec ma persévérance j'y suis arrivé. Et maintenant je suis libre ! 50% des issues BBB sont liés aux containers Docker, cherchez l'erreur...
RSSI et RGPD, voilà l'approche qui est arrivée assez rapidement; maîtriser les caméras, le son : Le consentement était donc LA solution. Encore une fois, je me suis chargé d'exprimer le besoin côté devs BBB (j'étais loin d'être le seul), mais un peu de forcing ça aide à faire bouger les choses. En quelques jours, c'était intégré de manière un peu forcé de ma part (sans annonce), mais au final, avec le recul, je sais que j'ai fait le bon choix !

Très vite j'ai du aussi me protéger des nos chers amis les moteurs de recherche qui commençaient gentiment à indexer certaines salles... Encore une fois, un peu de veille et quelque heure plus tard, la solution anti-référencement était en place. Aujourd'hui certains aspects liés à la sécurité de BBB sont encore en cours de développement (on se remet toujours en question).
Lorsque l'on parle de sécurité, on pense sécurité des données, intégrité des données, mais aussi et parfois on l'oublie, accessibilité des données. Très vite la solution scalelite a vu le jour pour permettre d'avoir un load balancing entre les instances BBB. Certains se sont lancés dans des infra avec du HA proxy ou des clusters nginx poussés. S'il y a bien un ennemis de la sécurité, c'est la complexité des applicatifs. Plus il y a de couches, plus les problèmes liés à la continuité de service sont difficiles à gérer/maîtriser. Mon expérience personnelle sur l'infra BBB : Ayant fait tomber intentionnellement ou involontairement les services en pleine production pour tenter de gérer une mini crise (il faut bien se mettre dans les conditions...), toujours mesurées, mais bon... , j'ai très vite compris ce qui allait me poser problème : La base redis gérant le scalling des instances BBB (une sorte de poubelle où les devs se donnent une joie de remplir...). Il fallait donc mettre en place un mode opératoire pour restaurer/recréer la base redis très rapidement (et surtout simplement). Pour le reste, hormis le pont audio via trunk SIP, pas de réel SPOF dans l'infra. Le frontend tombe ? On le reconstruit avec le snap de la veille. Opération assez rapide en soit. Après tout, nous ne sommes pas Google et leur 99,9999% de résilience.

- Aurais-tu un conseil pour les personnes qui souhaiteraient mettre en oeuvre une plateforme similaire dans leur établissement ?

Il faut se poser les bonnes question surtout : Pour qui ? Quelle population ? Moyen d'authentification à dispo ? Combien ? Quand (pensez au support et aux plages horaires sur lesquelles vous souhaitez offrir le service); je ne vous cache pas que ce genre d'outil ne peut pas être arrêté en plein milieu de journée, donc il faudra prévoir des HNO pour mettre à jour / tester... Avec quel moyen ? Humain tout d'abord, mais aussi machine. Il faudrait aussi éviter à mon sens de déployer n instances BBB dans chacun des labos. On ne peut décemment pas assurer la sécurité de toutes les instances que des E/C ont souhaité mettre en place dans leur coin. Prévoir donc une concertation pour savoir qui fait quoi et où. Penser à mettre très rapidement les AQSSI (directeurs d'établissement), à défaut le DPO (dans mon cas), pour faire adhérer à la solution. C'est tellement plus simple quand les ordres ou les lignes de conduites à suivre viennent d'en haut. Attention, un niveau de service à prévoir dans certain cas, sans pour autant parler de SLA : On est pas des prestataires privés ! Prévoir 2 semaines de prises en main complètes. 1 mois étant encore mieux. Il y a un peu de couche quand même à maîtriser, d'ailleurs ils ne s'en cachent pas : https://github.com/blindsidenetworks/scalelite :

" The Scalelite installation process requires advanced technical knowledge. You should, at a minimum, be very familar with
Setup and administration of a BigBlueButton server
Setup and administration of a Linux server and using common tools, such as systemd, to manage processes on the server
How the BigBlueButton API works with a front-end
How docker containers work
How UDP and TCP/IP work together
How to administrate a Linux Firewall
How to setup a TURN server
"

- Enfin, quels sont les points qui selon toi pourraient être améliorés dans BigBlueButton ou plus généralement sur les solutions de visio-conférences ?

J'ai revu un peu ma position sur ce point.
Doit-on parler de visio-conférence avec seulement des caméras, ou bien de web-conférence, qui s'apparente plus à un rassemblement de personne souhaitant suivre une conférence ?
Le point noir de BBB c'est sa capacité à encaisser la charge. J'ai bon espoir de monter à 500+ personnes dans un même salon au vu des modifications effectuées ces dernières semaines dans le code; prévoir tout de même je pense une machine physique en dur et pas un container pour ce type d'évènement.

BBB pourrait se doter d'un client lourd; sans doute que cela permettrait d'éviter les problèmes webrtc (liés principalement à l'OS / Le navigateur / le type de connexion / le firewall local...)
Le MENESR, fort de son succès avec bb-collab (private joke), a décidé d'investir dans BBB pour une solution nationale. Ce dernier ayant déjà investit dans la redistribution du code pour le pré-chargement des présentations à travers Greenlight. Je pense donc avoir fait le bon choix d'approfondir cette solution. Une certification ANSSI en vue ?
Pour avoir essayé jitsi-meet qui semble plus adaptée pour une visio en petit comité, BBB reste aujourd'hui une solution "abordable". Prévoir tout de même comme évoqué, une veille pour toutes les technos dont la solution dépend.
Si on combine Zimbra, Moodle, rocket.chat, nextcloud, esup portail et BBB; le tout avec une fédération d'identité en frontal, on se retrouve avec une solution opensource où toutes les API s'imbriquent les unes aux autres et on dispose alors d'une infra pouvant débouter un Microsoft Teams. Ce n'est qu'une piste de réflexion.

Merci Aurélien pour ce précieux retour d'expérience ! Article rédigé par Aurélien Guerson, Olivier Levillain et Patrick Bas, Contact : aurelien.guerson@imtbs-tsp.eu