La sécurité des systèmes mainframes

Stéphane Diacquenod
stephane.diacquenod@volvo.com

Volvo IT

Résumé Souvent au coeur des SI de grands groupes (banque, industrie, etc.), le mainframe reste méconnu des RSSI et des équipes techniques des autres plateformes. Cet article propose donc d'évaluer avec un oeil objectif leur sécurité et de présenter leur spécificité afin de mieux le prendre en compte dans la sécurité des SI.

1 Introduction

Le mainframe est parfois perçu comme une technologie désuète, beaucoup avaient même envisagé sa disparition avec le passage à l'an 2000 et son fameux bug. Pourtant le mainframe est toujours présent et le nombre de mainframes serait même en augmentation d'après IBM. Même s'il est difficile d'obtenir des chiffres à jour fiables, il est certain que le mainframe est au coeur de nombreux systèmes critiques.

Les utilisations sont systématiques dans l'industrie bancaire : gestion des comptes et système d'autorisation de carte de crédit utilisent presque exclusivement le mainframe.

Dans les domaines de l'assurance ou du service, ils sont fréquemment utilisés pour la gestion des contrats et pour les bases de données clients.

Dans l'industrie lourde et la distribution, il permet la gestion des comptes, des stocks et des commandes. Dans certaines usines, il assure aussi la coordination des flux et donc l'ordonnancement des chaines de montage.

Installé dans ces entreprises au début de l'informatisation, on pourrait penser qu'il y est resté en place du fait de sa criticité ou pour des difficultés de migration, mais on le trouve aussi dans des systèmes d'information plus récents (année 90).

En fait, vous utilisez tous les jours des dizaines de systèmes mainframe différents sans même en avoir conscience.

Fêtant ses 50 ans cette année, le mainframe a vu changer son paysage technologique. L'arrivée des réseaux IP l'a même connecté au reste du monde en l'exposant, au passage, à de nouveaux risques.

La question de la sécurité de cette plateforme n'est donc pas anodine et il est important de voir si le mainframe mérite toute la confiance qu'on lui accorde.

2 Architecture des systèmes mainframe

2.1 Introduction

Le mainframe est aujourd'hui un système informatique hybride. IBM a, au fil des années, rajouté des composants matériels à son architecture mainframe historique. Le dernier né de chez IBM, sorti en 2013, est officiellement nommé zEntreprise EC12. Il est composé de plusieurs éléments :

- Un ou plusieurs zBX, ensemble de serveurs basés sur l'architecture Power d'IBM et sur le classique x86. Ce composant reste encore très peu utilisé.
- Le Central Processor Complex (CPC), contenant les processeurs basés sur l'architecture z, c'est le mainframe à proprement parler.
- Une ou plusieurs baies de disques.
- Une ou plusieurs robotiques (physiques ou virtuelles).

La politique actuelle d'IBM semble être de mettre en place une architecture hybride afin d'obtenir le meilleur de chaque architecture. Le zBX était une première approche dans ce sens avec l'ajout d'une baie pour recevoir des processeurs Power. La seconde version permet la mise en place de lames x86 ouvrant ainsi la voie aux machines virtuelles Windows.

Les clients n'ont pas été réceptifs à cette approche, mais les nouvelles extensions couplant un matériel et un logiciel spécialisé pour le traitement de certaines requêtes DB2 (Système IDAA) sont beaucoup plus attendues. Nous nous intéresserons ici uniquement à la partie mainframe "historique" et donc à la CPC.

2.2 Mainframe

Le CPC est ordinateur complet composé de :

- l'alimentation électrique ;
- le système de refroidissement à eau ;
- les connexions pour les cartes d'entrée / sortie ;
- les Supports Elements (SE), PC de gestion du mainframe ;
- les 4 Flexible Support Processeur (FSP).

Chaque FSP contient 1 module de refroidissement (MCM) avec 6 processeurs et 2 contrôleurs de mémoire. Au module de refroidissement est adossé un maximum de 768 Go de RAM. Le mainframe peut proposer

jusqu'à 101 PU (processor unit, coeur des processeurs) et 3 To de RAM pour l'ensemble des systèmes. Les PU peuvent être de plusieurs types :

- Central Processor (CP) chargé des travaux classiques ;
- System Assistant Processor (SAP) chargé des I/O ;
- Internal Coupling Facility (ICF) chargé de la gestion des synchronisations en cas de haute disponibilité (parallele sysplex) ;
- z Integrated Information Processor (zIIP) qui décharge le CP de certain flux dont XML et Java ;
- Internal Facility for Linux (IFL) qui fait tourner les systèmes zLinux.

Il est possible de conserver des processeurs supplémentaires afin de prendre la place d'un éventuel processeur défaillant. Il est aussi possible de répartir ces différents processeurs dans les différents FSP afin de garantir la disponibilité en cas de perte ou de la maintenance de l'un des FSP.

2.3 Processeurs

Le mainframe dispose de processeurs hexacoeurs opérant à 5,5GHz. Chaque coeur possédant ces mémoires L1 et L2 dédiées, la mémoire L3 est partagée entre les coeurs d'un processeur. Chaque module de refroidissement (FSP) dispose de 2 contrôleurs mémoire et une L4 partagée entre les 6 processeurs. Chaque cœur (PU) dispose d'une unité d'accélération matérielle appelée CPACF. Cette unité propose des accélérations matérielles aux opérations de chiffrement et de cryptographie. CPACF permet de décharger l'unité de calcul classique de certaines opérations cryptographiques telles que DES, TDES, AES, PRNG, MAC, SHA1, SHA2.

2.4 Crypto-processeur

Le mainframe dispose aussi d'un coprocesseur cryptographique. Cette carte peut être utilisée comme coprocesseur ou comme HSM. Dans ce dernier cas, il est nécessaire de mettre en place une station de saisie des clés (TKE) afin de saisir les clés cryptographiques utilisées. Cette machine est elle-même munie d'un HSM. Les HSM du mainframe sont accessibles par PKCS11 ou par CAA (un standard IBM) ; dans les deux modes il permet d'utiliser les algorithmes standards (DES, TDES, AES, RSA, SHA1, SHA2). On peut regretter qu'IBM n'ait pas retiré le support d'algorithme comme DES et TDES. Aucune annonce de leur retrait n'a été réalisée pour l'instant.

2.5 Hyperviseur

Le mainframe fonctionne avec un hyperviseur de type 1 nommé PR/SM (Processor Ressource / System Manager). Cet hyperviseur ne peut être manipulé que par IBM. L'hyperviseur assure un mécanisme de partition logique (LPAR) afin de séparer les divers environnements clients. Les LPAR assurent une étanchéité certifiée EAL5 entre ceux-ci. Un mainframe peut actuellement faire tourner 60 LPAR maximum. Chaque LPAR disposant de son propre système d'exploitation supporté par le mainframe :

z/OS : le système le plus utilisé ;

z/VSE : l'évolution des systèmes DOS ;

z/TPF : l'OS temps réel ;

z/Linux : le meilleur des deux mondes :) ;

z/VM : le système de machine virtuelle pour mainframe. Chaque VM peut faire tourner son OS parmi ceux précités.

L'environnement z/OS représenterait entre 70 à 80 % des systèmes mainframe. z/VSE quant à lui se fait plus rare et semble en perte de vitesse et enfin z/TPF est un segment ultra spécialisé (quelque machines dans le monde, semble-t-il). Nous ne parlerons que de z/OS dans la suite de la présentation.

3 Le système z/OS

3.1 Vue d'ensemble

z/OS est le système d'exploitation de référence des plateformes mainframe. Il est issu de la longue lignée des systèmes MVS exploités depuis 1964. Ce système a toujours été perçu comme robuste, ciblant les applications hautement disponibles et à forte sollicitation. Certains composants furent, à une époque lointaine, diffusés sous forme de code source, mais la norme est, depuis bien longtemps, la diffusion de modules binaires compilés pour l'architecture mainframe. Les travaux d'analyse du code de z/OS sont rares (voire inexistants) et non diffusés publiquement. Si l'on se fie à l'expérience des administrateurs de cette plateforme, de nombreux modules ne sont pas modifiés lors des mises à jour. En effet, IBM semble adepte de l'ajout de nouveaux modules, chacun réalisant sa tâche, ce qui permet de ne pas introduire de nouveau bug dans des modules matures. On retrouve là le principe de simplicité des programmes d'Unix. Contrairement aux idées reçues, la qualité du code de z/OS ne semble pas supérieure

à celle d'un Unix classique. Des correctifs mineurs (PTF) sont apportés régulièrement. La stabilité du système est malgré tout rarement remise en cause par ces bugs, fonctionnalité mineure ou performance constituant la majeure partie des PTF. Force est aussi de constater que les bugs affectant l'intégrité des données sont rarissimes. Ces PTFs sont publiées par IBM sous le terme « Red Alert ». IBM et les autres éditeurs ne semblent pas disposer de système de suivi des vulnérabilités pour les produits mainframe. Ce manque de suivi s'explique par l'extrême rareté des codes malveillants sur mainframe.

3.2 Instructions privilégiées

Le jeu d'instruction assembleur du mainframe protège certaines instructions (privilégiée ou semi-privilégiée). Pour pouvoir exécuter ces instructions, le processeur doit être préalablement placé en mode superviseur. Le fonctionnement est similaire aux rings de l'architecture x86. La macro qui bascule le processeur de mode nominal (dit mode problème) au mode superviseur nécessite que :

- le module ait été compilé avec l'option AC=1 par le développeur.
- la librairie contenant le module ait été ajoutée à la liste des librairies autorisées par l'administrateur.

Les portions de code nécessitant le mode superviseur peuvent alors être réduites à leur minimum et clairement cloisonnées, diminuant ainsi la surface d'attaque. La liste des programmes autorisés est l'APF, elle, peut être modifiée dynamiquement par les administrateurs autorisés. L'APF est un point très important de la sécurité des mainframes, il est crucial que les changements dans cette liste soient suivis par les équipes système. C'est en règle générale le cas.

3.3 Protection de la mémoire

z/OS alloue des pages mémoires de 4 ko, chaque page est protégée par une clé codée sur 8 bits. Les programmes ne peuvent pas utiliser une page si leur clé ne correspond pas à celle de la page. Les clés sont généralement associées comme suit :

Key	Type de programme autorisé	Exemple
0	accès superviseur	Nucléus
1	Interface système	JES, TSO , planificateur de job
2	Réservé	
3	Haute Disponibilité	Availability management (AVM)
4	Réservé	
5	Gestionnaire d'E/S	SMS, RMM
6	Réseaux	VTAM, IP
7	Base de données	CICS, DB2, MQ
8	Mémoire virtuel	Programme utilisateur (défaut)
9		Programme utilisateur
A-F	Memoire réel	Programme utilisateur

Par défaut, les programmes utilisateurs fonctionnent avec la clé 8. Pour disposer d'une autre clé, il faudra réaliser une définition dans la « Program Properties Tables » (PPT). Cette définition est statique. La clé est complétée par un bit « fetch » qui définit si la page est accessible en lecture publique. On obtient alors le tableau d'accès suivant :

Fetch bit	Key	Accès en lecture	Accès en écriture
0	Égale	Oui	Oui
0	Non égale	Oui	Non
1	Égale	Oui	Oui
1	Non égale	Non	Non
.	0	Oui	Oui

La protection entre les programmes utilisateurs, tous en clé 8 est assurée par le système lui-même lors de l'affectation des pages. Sans être parfait, ce système offre une protection des pages système.

3.4 Unix system services

Le mainframe contient une implémentation d'un système UNIX tournant dans le z/OS. Cette implémentation nommée USS (ou OMVS) est loin d'être une lubie. C'est un composant important du mainframe qui permet le fonctionnement de l'ensemble des applications utilisant TCP/IP. Seulement voilà, les administrateurs mainframe sont... des administrateurs mainframe. Leur connaissance des systèmes UNIX est souvent limitée ou inexistante et cela peut générer de gros problèmes de sécurité. Il est donc essentiel d'assurer la protection des fichiers UNIX comme celle des

fichiers z/OS. Il faut définir tous les utilisateurs ayant accès à OMVS correctement, avec les droits, le home et l'UID qui correspondent. On peut aussi prendre le parti d'interdire OMVS au plus grand nombre. Il est également intéressant de fixer des règles claires sur l'emplacement des fichiers de configuration lorsque ceux-ci peuvent être dans z/OS ou dans OMVS.

3.5 Aspect réseau

IBM a conçu en 1974 un modèle de réseau multiplateforme nommé SNA (System Network Architecture). Son implémentation sur mainframe se réalise à travers VTAM qui est (et restera) un composant central du système. Ce réseau permet de connecter les différentes applications entre elles à travers des LU (Logical Unit) comparable à des sockets. Ces LU permettent aussi la connexion de terminaux utilisateurs. On peut aussi présenter des ressources VTAM d'une machine sur une autre machine à travers une liaison IP grâce au protocole Entreprise Extender (RFC 2353). La liaison IP est alors réalisée de mainframe à mainframe. Du fait de cette architecture, la sécurité n'est pas prise en compte alors que les données transitent en clair (5 ports UDP, en fonction de la priorité des messages). Entreprise Extender peut être encapsulé dans IPsec ou être configuré pour disposer de SNA SLE (Session Level Encryption). SNA SLE souffre du même problème qu'IPsec (que ce soit dans le monde mainframe ou non). Leur mise en place est difficile et souvent repoussée du fait du temps nécéssaire à sa mise en place.

Les connexions des clients sont, elles, réalisées grâce au protocole Telnet 3270 (TN3270) qui n'est qu'un flux Telnet enrichi sémantiquement. L'ensemble des données est ainsi transmis en clair sur le réseau. Bien qu'une encapsulation dans TLS soit possible, les mises en place ne sont pas systématiques.

La problématique de redondance des connexions réseau, habituellement résolue en niveau 2 par ARP spoofing, est ici gérée en niveau 3 par routage dynamique. En effet, les machines ont en général 2 cartes réseau disposant chacune de leur propre adresse IP. Un démon de routage OSPF est mis en oeuvre sur la machine et va annoncer la ou les adresses virtuelles utilisées par le système. On trouve régulièrement des machines disposant de plusieurs piles IP (oui, IP est un produit comme un autre que l'on peut arrêter). En règle générale, il y aura un démon de routage par pile IP.

4 Contrôle d'accès

4.1 System Access Facility

SAF est une API de sécurité de z/OS, profondément ancrée dans le système et dont l'appel ne peut pas être contourné. SAF sera appelée à chaque fois que l'on souhaite accéder à une ressource. Les ressources seront essentiellement les fichiers (appelés dataset), mais aussi les commandes système (commande ISPF) ou les accès au log (syslog ou joblog), SAF permet d'interfacer un outil de sécurité avec le système d'exploitation. Il existe aujourd'hui 3 produits sur le marché :

- RACF (Resource Access Control Facility) d'IBM ;
- ACF2 (Access Control Facility) de CA ;
- TSS (TopSecret Security) de CA.

Le produit de référence est RACF (car IBM), ACF2 étant de moins en moins utilisé car il met en oeuvre une logique très particulière. Ces produits de sécurité permettent d'assurer l'identification des utilisateurs, le contrôle des accès et dans une certaine mesure, l'enregistrement des actions. L'identification des utilisateurs peut être effectuée grâce à une source LDAP externe. De plus, chaque produit peut venir avec ses propres définitions pour étendre ou ajouter des ressources à protéger. La sécurité de tous les produits peut (et devrait) être assurée par RACF.

4.2 Mot de passe

Comme tous les outils de sécurité, RACF stocke des données permettant d'identifier les utilisateurs. Une simple commande permet de déterminer l'emplacement des bases RACF. Malheureusement, RACF ne stocke pas les mots de passe d'une manière satisfaisante, il est donc important de protéger ces fichiers même en simple lecture. RACF chiffre le nom d'utilisateur avec un dérivé du mot de passe grâce à l'algorithme DES puis stocke le résultat en base.

Le mot de passe est traditionnellement limité à 8 caractères comprenant majuscule, minuscule et chiffre. Par défaut, le champ mot de passe n'est pas sensible à la casse. RACF stocke plusieurs anciens mots de passe pour contrôler la non-répétition de ceux-ci. Les utilisateurs sont révoqués après quelques tentatives infructueuses (entre 3 et 20 selon les configurations). Une fonction de phrase est disponible pour dépasser la limite des 8 caractères, mais elle n'est pas supportée par certains applicatifs.

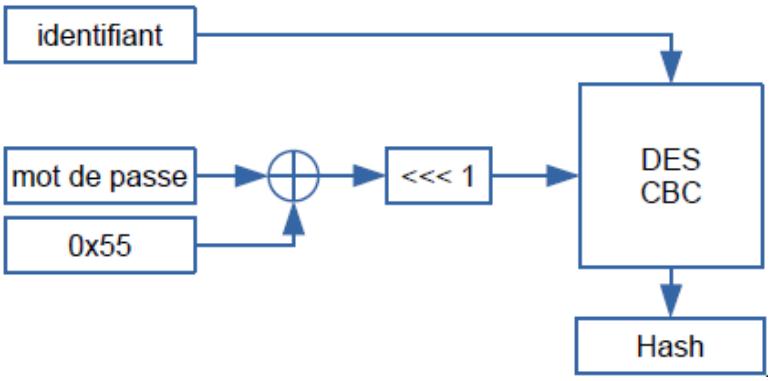

Figure 1. Algorithme de stockage des mots de passe sous RACF

4.3 Autorisation

Les règles de protection d'une ressource peuvent être appliquées à des groupes ou à des utilisateurs. Il est bien évidemment préférable de définir les règles sur des groupes auxquels on connectera les utilisateurs concernés. Cependant, certains droits sont associés aux utilisateurs et viennent surcharger les règles d'autorisation spécifiées précédemment.

Compte spécial : Le compte spécial est administrateur de la base RACF. Il peut, à convenance, créer des règles, des groupes ou des utilisateurs dans la base des autorisations. En revanche, son compte ne dispose pas de droits particuliers sur les ressources (pas d'accès universel notamment).

Compte opération : Ce droit permet aux operateurs d'effectuer les operations de gestion quotidienne des fichiers de production sans avoir à donner des droits explicites sur les fichiers. Il permet par exemple de déplacer les fichiers d'un disque à l'autre ou encore d'effectuer la mise sur bande. Ces opérations sont aujourd'hui effectuées par des logiciels de gestion automatique de l'espace de stockage.

Compte audit : L'auditeur peut accéder en lecture à de très nombreuses ressources à des fins de contrôle de la sécurité (SMF, profile RACF, etc.). Son rôle premier est de contrôler l'activité des comptes disposant de l'attribut Special.

Ces attributs peuvent être définis au niveau d'un groupe ou du système complet. Un des paramètres importants de RACF est sa configuration d'accès par défaut. Nous savons tous que la bonne méthode est d'interdire les accès par défaut, puis d'autoriser les accès explicitement pour chaque groupe. Dans ce cas, les utilisateurs ne disposant pas de règle d'accès

les concernant seront rejetés. Ce n'est pas la configuration par défaut de RACF (option NOPROTECTALL).

RACF dispose aussi d'un mécanisme de SECLABEL/SECLEVEL qui permet des contrôles supplémentaires en fonction de la criticité de l'information souhaitée. Ce système vise essentiellement les SI où la criticité de l'information est clairement catégorisée (milieux militaires par exemple).

4.4 Protection des fichiers

Les fichiers représentant la majorité des ressources à protéger, la difficulté de la protection des fichiers sous mainframe réside dans le fait qu'aucune norme de fichier n'est préétablie. Certaines ressources sont codifiées par IBM car elles sont utilisées très tôt dans le processus de démarrage (IPL), mais elles sont très peu nombreuses. Il appartient donc à l'administrateur système de définir une norme de fichier et de la maintenir. Cette norme doit permettre :

- de différencier les divers environnements si ceux-ci sont sur la même machine (production, recette, qualification, développement, bac à sable, etc.)
- de différencier les divers groupes d'utilisateurs (système, exploitation, développement, utilisateur, etc.)

Une simple matrice entre ces deux contraintes doit permettre de couvrir la quasi-totalité des usages.

4.5 TSO, ISPF, SDSF

TSO est une sorte de shell (personne ne l'utilise tel quel), ISPF vient l'habiller d'une interface où l'on navigue par panneaux ou écran successifs et dans lesquels les différents outils sont disponibles. SDSF est un des outils présents dans ISPF, il permet de s'interfacer avec le système z/OS et son interpréteur de job, JES. L'ensemble de ces outils constitue l'environnement de travail d'un utilisateur mainframe. ISPF ne dispose pas de règle de protection à proprement parler. Permettant principalement de naviguer parmi les fichiers, les règles applicables sont celles de l'accès au fichier défini précédemment.

SDSF est un composant plus intéressant, car il permet d'aller contrôler les files d'attente du gestionnaire de job et de lire leur résultat d'exécution. Il permet aussi la consultation de la log système et l'exécution des commandes. Toutes ces ressources (files d'attente JES ou commande SDSF) peuvent être protégées par RACF. Les règles de protection fonction/environnement

semblent là aussi pertinentes dans la majorité des cas. Elles ne sont pas souvent mises en place de façon satisfaisante.

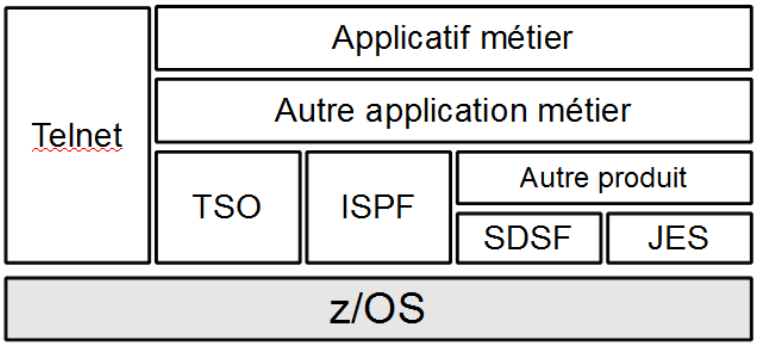

Figure 2. Vue d'ensemble d'un environnement mainframe

4.6 Contrôle des STC

Lors du démarrage d'une STC, JES fait appel à RACF pour attribuer un utilisateur à la STC. Chaque STC doit disposer d'un utilisateur avec les droits nécessaires. Protéger les fichiers utilisés par la STC contre l'accès et la modification par d'autres utilisateurs (techniques ou réels) est indispensable pour garantir la sécurité.

Un exit de RACF (ICHRINE03) est appellé en amont du contrôle de droit dans RACF. Il est indispensable que cet exit ne donne aucun droit et que tous les droits soient définis dans RACF.

4.7 Server FTP

Le server FTP du mainframe dispose d'une fonctionnalité très intéressante. En effet des commandes « SITE... » permettent de préciser la structure du fichier que l'on va créer sur le mainframe par exemple. Une des commandes site est SITE FILETYPE=JES. Cette commande envoie le fichier transféré directement dans la file d'entrée de JES. Cette fonctionnalité est très décriée, car elle permet de contourner la politique générale de sécurité et on peut trouver des exemples d'attaque utilisant cette fonctionnalité sur internet. Cependant, on oublie souvent de préciser que ceci n'est possible que sous certaines conditions. Pour commencer, il est nécessaire de disposer d'un compte/mot de passe valide sur le mainframe avec des droits de connexion TSO, à moins que la connexion anonyme n'ait été activée. Ensuite, l'option JESINTERFACELEVEL 2 permet de

protéger les jobs soumis à JES par RACF, ce qui revient à appliquer des politiques de contrôle d'une soumission de job classique. Sur le même modèle, il est possible de réaliser des requêtes DB2.

4.8 Global Access Checking

La GAC (Global Access Checking) est une table d'autorisation très particulière. Créée pour améliorer les performances, elle permet de créer des règles de sécurité en dehors de RACF. Cette table est appelée avant RACF et son parcours est interrompu ausitôt qu'une règle correspond. RACF n'est interrogé que si aucune règle ne correspond.

Le risque de sécurité est évident. Il est indispensable que la GAC soit sous le contrôle des administrateurs sécurité. Tout ajout de règle doit se faire avec une grande minutie et de grandes précautions car les effets sur la sécurité peuvent être dramatiques.

4.9 Exit RACF, suite et fin

RACF dispose de nombreux exit (comme beaucoup de produits mainframe). Deux d'entre eux méritent une petite attention. Ces exit (ICHDEX01 et ICHDEX11) permettent de modifier le comportement de RACF sur le stockage et la vérification des mots de passe. Ils peuvent donc être utilisés pour améliorer la sécurité, en choisissant sa propre méthode de stockage de mot de passe mais auccun retour n'a pu être trouvé sur ce genre de pratique. Ils peuvent aussi être utilisés pour que RACF réponde positivement à toutes les demandes d'autentification d'un utilisateur. L'emplacement de ce module nécessite néanmoins que l'utilisateur dispose déjà de droits très avancés sur le système pour réaliser ce genre d'opération.

5 Journalisation

Tout Job ou STC démarré dispose de ses propres log, lui permettant de tracer son exécution ainsi que toutes les anomalies rencontrées. En parallèle de cela, d'autres mécanismes sont utilisés.

5.1 Trace d'exécution de programme

Tous les programmes (Jobs, STCs...) disposent d'une trace d'exécution, très riche en information. On y retrouve le code JCL exécuté, tous les fichiers utilisés par le système, les messages d'exécution, d'information ou d'erreurs rencontrées lors de l'exécution du programme. Des règles de

gestion permettent de définir la durée et les modalités de rétention de ces données.

5.2 Trace d'exécution système

La trace d'exécution du système (syslog), un peu à la façon de la syslog Unix, trace toutes les exécutions de programme, début et fin de job ou de STC, les connexions d'utilisateurs, ainsi que les messages à destination des opérateurs (WTO, WTOR).

5.3 Trace des erreurs

La trace des erreurs (logrec) était à l'origine utilisée plutôt par les équipements (baie disque, robotique, et autres). On y trouve aujourd'hui toutes les erreurs, matérielles et logicielles, nécessitant l'attention de l'administrateur. Contrairement à syslog, aucune information d'exécution n'est stockée ici, mais uniquement des erreurs.

5.4 SMF

Le System Management Facilities est un système d'enregistrement de l'activité. Les administrateurs peuvent sélectionner les enregistrements à conserver ou non. Les enregistrements SMF sont très utilisés pour effectuer des analyses de performance avec les enregistrements suivants :

Type	Information
30	Consommation de chaque Job ou STC
70-79	Consommation du système
100-102	Performance de DB2
110	Performance de CICS
115-116	Performance de MQ

Des enregistrements spécifiques peuvent aussi être utilisés pour assurer la traçabilité des modifications effectuées sur un système, par exemple :

Type	Information
80-81	Autorisation RACF
64 14-15	Accès à un data set

Les enregistrements SMF sont très nombreux et très détaillés, il parait impossible qu'une action puisse avoir lieu sur le système sans être enregistrée dans un enregistrement SMF. L'exploitation de ces données peut par contre être très difficile, car il est nécessaire d'écrire des parsers pour extraire ces données.

5.5 Traitement et conservation

Les traces des jobs ou du système peuvent être traitées par des outils spécialisés qui s'occuperont de conserver les informations selon les politiques définies (durée, support, etc.). Il est même possible de créer des outils non modifiables par l'utilisateur. Les logs se retrouvent ainsi protégés des modification manuelles, permettant ainsi de garantir leur intégrités lorsque cela est nécessaire (tâche liée au traitement bancaire ou à la cryptographie, log RACF...).

Comme toujours, il est très important de définir quelle trace doit être conservée et pour quelle durée. Le problème du traitement et de la collecte de ces données est très important du fait de la diversité des sources et de la volumétrie. Les logs de job et système sont textuels, mais les logrec et les enregistrements SMF sont des formats binaires. De plus, chaque type d'enregistrement SMF dispose de son propre format. Les développements des outils d'analyse de ces enregistrements peuvent rapidement devenir très complexes et très coûteux.

6 Vue applicative

6.1 Architecture applicative traditionelle

Une application mainframe traditionnelle fonctionne sur le duo CICS/DB2 (on trouve aussi de l'IMS). Elle sera accessible à travers une interface en mode texte et l'utilisateur naviguera dans les écrans avec les touches de fonction (F1 à F24). Dans ce cas, on accède à l'application avec le protocole Telnet 3270. La partie interactive est en général complétée par un traitement par lot de nuit (batch).

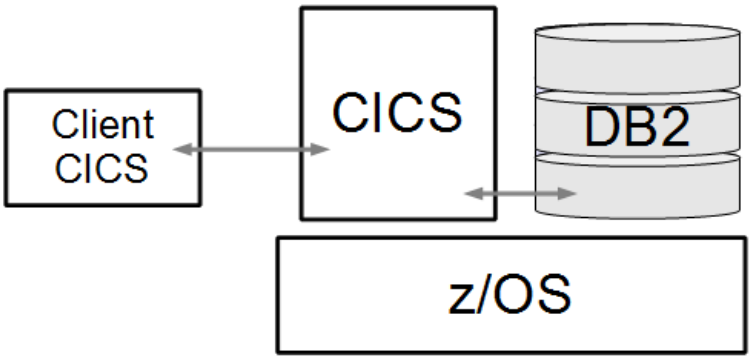

Figure 3. Architecture traditionnelle d'une application mainframe

Ces applications sont généralement écrites en Cobol. Les ordres SQL d'DB2 mainframe ont la particularité d'être précompilés pour des raisons de performance (pas de recalcul du chemin d'accès). Ceci est aussi un gain pour la sécurité car il n'est pas possible de réaliser un injection de code SQL.

Dans certains applicatifs, aux requêtes SQL difficiles à construire, on réalise des programmes utilisant des requêtes SQL dynamiques. L'injection SQL peut donc sembler envisageable mais cobol construit les ordres SQL en deux temps, préparation de l'ordre puis insersion des variables. Cette préparation des requêtes, couplée avec une vérification des entrées utilisateurs, permet de se prémunir des injections SQL.

Dans cette architecture applicative, un équivalent à des attaques XSS ou CSRF n'est pas envisageable, le client ne disposant que d'une capacité d'affichage des données envoyées.

S'il semble donc difficile d'attaquer l'application à proprement parler, il est trivial d'attaquer le protocole applicatif (Telnet) car il ne propose aucune protection par défaut.

6.2 Architecture applicative modernisée

Avec le développement des réseaux, les applications sont de plus en plus interactives. Certains composants de l'application, traités par le passé par un batch de nuit, ont été convertis pour permettre l'utilisation interactive. Cela modifie les courbes de consommation des applications et certains clients ont maintenant 80 % de leur consommation sur leur applicatif interactif (contre 40 à 60 % dans une application traditionnelle). Les interfaces sont aussi modernisées, les écrans noirs et verts sont de plus en plus remplacés par des interfaces web. Plusieurs stratégies peuvent ici être abordées :

- habillage d'écran ;
- génération des pages HTML sur mainframe ;
- architecture de web service.

Sur le fond de l'application, rien ne change, nous exécutons toujours du code serveur pour générer les écrans et les requêtes SQL sont toujours précompilées.

La génération de page web avec du JavaScript peut ajouter des risques classiques et connus des gestionnaires d'application web (mais pas par des mainframeurs). Il est nécessaire de noter que ces risques étant nouveaux pour les équipes en place, elles ne possèdent pas nécéssairement les connaissances nécéssaires à leur gestion.

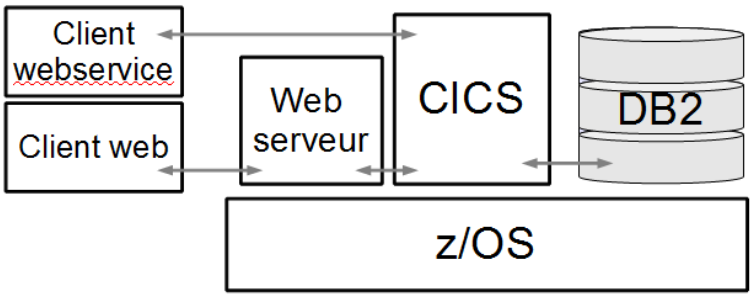

Figure 4. Exemple d'architecture d'une application modernisée

Force est de constater que malgré le changement d'architecture de l'application et quel que soit le mode de transformation choisi, la vulnérabilité la plus courante reste la même.

Les pages ou les webservices sont rarement servis à travers TLS, que ce soit à destination des utilisateurs finaux ou des applications intermédiaires.

7 Outils d'audit

Les outils d'audit classiques de sécurité ne semblent pas prendre en compte les systèmes mainframes. Ceux-ci sont en générale scrupuleusement mis de côté par les équipes en charge du suivi de la sécurité. Voici un rapide aperçu des capacités des quelques scanners de vulnérabilité :

Outils	Test pour environnement Mainframe
Nessus	0
OpenVAS	0
McAfee	0

Ici, avec NMAP, la détection d'OS est complètement erronée, nous ne nous trouvons même pas dans la bonne famille d'OS. NMAP est induit en erreur à cause du nombre important de services fonctionnant dans les USS. Ceux-ci sont souvent des versions modifiées de programme libre pour Linux, ils ont donc une signature semblable.

Depuis peu, les outils d'audit libres sont patchés afin de prendre en compte le mainframe. Ceci est notamment dû au travail fourni par « Soldier of Fortran ». Ce dernier a par exemple travaillé sur la prise en charge par John the Ripper des bases RACF.

Il a aussi développé un meterpreteur à la façon de metasploit pour l'environnement z/OS (qui prend en compte les USS).

Ces outils permettent de mieux prendre en compte les mainframes dans les politiques d'audit. Mais faire utiliser ces outils par les administrateurs mainframe est une mission délicate, voire impossible.

Il faut aussi prendre en compte les outils d'audit spécifiques au mainframe, par exemple le health checker de z/OS ou les capacités d'audit de RACF permettent de contrôler de nombreux points d'une politique de sécurité.

Comme les outils classiques sont inopérants sur les systèmes mainframes, une collaboration étroite entre les équipes des RSSI et les équipes de sécurité mainframe est nécessaire, plus encore que pour les autres platformes.

8 Conclusion

Quel niveau de sécurité est capable d'assurer le mainframe ?

Nous avons vu au cours de cet article que de nombreux outils sont présents pour assurer la sécurité de la plateforme. Leurs niveaux de sécurité sont homogènes et peuvent permettre un très haut niveau de sécurité. La mise en oeuvre de ces outils, parfois non actifs par défaut, est de la responsabilité des équipes techniques. Il apparait alors que le niveau de sécurité des systèmes peut être assez variable en fonction de l'attention et de l'engagement des équipes, techniques et managériales, vis-à-vis des problématiques de sécurité.

De ce fait, il est important que l'ensemble des intervenants comprenne les risques auxquels le système est exposé. Cette compréhension est un élément clé pour cette plateforme, car les risques ont fortement évolué durant les 50 ans de vie du mainframe.

Depuis sa naissance en 1964 jusque dans les années 90-95, le mainframe à vécu en autarcie, ne communiquant qu'à travers son réseau SNA avec des terminaux 3270. Mais depuis les années 90, le mainframe s'est ouvert sur le monde, à travers le réseau IP, l'exposant ainsi à de nouveaux risques.

Ce changement majeur de technologie trouve un écho dans le changement de génération. On observe en effet dans de nombreuses équipes une répartition des âges très déséquilibrée et un fort besoin de sang neuf.

L'enjeu étant donc d'apporter de nouveaux techniciens dans ces équipes, cela doit permettre de réaliser les transferts de compétence nécessaires et de rééquilibrer l'échelle des âges. C'est aussi une opportunité pour insérer de nouvelles cultures techniques dans les équipes, issues du monde IP et des systèmes Linux.

IBM cherche d'ailleurs à encourager ce renouvèlement et met en place une stratégie de communication importante pour rendre sa plateforme

attractive pour les administrateurs, les développeurs et les décideurs. Ces efforts de communication sont cohérents avec les évolutions techniques de la plateforme.

Sur le plan de l'administration, z/Linux est mis en avant afin de devenir le système phare du System/z. Sur z/OS, les nouveaux types de charges de travail sont mise en avant avec des tarifications réduites pour le Java, les travaux XML et la gestion des flux IP. Les outils de gestion graphique (z/OSMF) viennent compléter les terminaux verts et noirs chers à nos coeurs pour offrir des vues cohérentes et synthétiques du système. Cela permet aussi, à de nouveaux arrivants, d'apprivoiser plus facilement les outils mainframe.

Les développeurs disposent maintenant d'outils déportés sur les postes clients simplifiant ainsi l'accès au mainframe grâce à l'utilisation d'outils classiques (eclipse, SVN...). Des outils équivalents permettent de simplifier l'accès à la gestion des bases de données

Les capacités de scalabilité linéaire et de big data du mainframe sont promues par IBM auprès des décideurs. Ces actions viennent parfaire une gamme déjà très complète avec z/Linux, les zBX et les différentes appliances spécialisées fournies par IBM pour leur plateforme phare.

Le mainframe, après un demi-siècle de service, a donc su s'adapter. Il dispose aujourd'hui d'atouts majeurs pour faire face à la concurrence sur les marchés porteurs du Cloud et du big data. IBM, en vendant toute sa gamme PC Intel (client et server) à Lenovo, a fait le pari des serveurs à haute valeur ajoutée, le mainframe est le fer de lance de cette politique. Prenons donc rendez-vous le 8 Avril 2024 pour fêter ses 60 ans et voir quelle place Big Blue a réussi à lui donner dans le paysage de l'IT.

Références

1. Philip Emrich. Top ten z/os & racf audit findings. `www.fspgroup.ca/docs/FSP20081003_2.pdf`, 2013.
2. IBM. Security server racf auditor's guide. SA22-7684-13, 2013.
3. IBM. System management facilities. SA22-7630-26, 2013.
4. Soldier of Fortran. Blog. `http://mainframed767.tumblr.com/`.
5. Wihelm G. Spruth. Initiation au mainframe. `www.informatik.uni-leipzig.de/cs/Literature/Features/report.pdf`, 2008.