Normes d'orientation du secteur pour les tiers en matière de « Encodage sécurisé »

Émission finale 1.0

Date : 19/10/2016,

Sécurité de BT

Tables des matières

1. Introduction 3
1.1. Contexte 3
1.2. Objectifs 3
1.3. Matériel supplémentaire et conseil 4
2. Conception du système 5
3. Fourniture du système 13
4. Assurance du système 19
5. Annexes 21
5.1. Contrôle de l'accès 21
5.2 Cryptographie 23
5.2.1 Contrôles de cryptographie web 25
5.2.2 Gestion générale des clés 26
Contrôle du document 28

1. Introduction

1.1. Contexte

Ce document expose les directives générales sur les bonnes pratiques pour le développement de logiciel réalisé par des tiers qui fournissent des applications ou des systèmes à BT.

Les vulnérabilités de la couche application permettent aux personnes d'avoir très facilement accès aux données ou aux systèmes. Il est donc de plus en plus important que le logiciel intègre des aspects de sécurité en son noyau dans son développement.

1.2. Objectifs

Ce document vise à fournir un ensemble d'orientations sur la façon de développer le software de sorte que le code qui en résulte soit fiable et sûr. Le projet de développement d'un logiciel revêt de nombreux aspects et c'est un domaine qui évolue rapidement, c'est pourquoi nous avons ébauché, dans la mesure du possible, un ensemble de principes de haut niveau pour les différentes étapes du projet de logiciel. Le document est divisé en trois sections principales :

Conception du système

Cette section donne des conseils pour le démarrage d'un projet de logiciel et souligne divers aspects qu'il faudra prendre en considération avant d'écrire tout code. En suivant ces étapes, il sera plus facile plus tard, durant la vie utile du projet, d'assurer la délivrance d'un code sécurisé.

Fourniture du système

Cette section couvre le développement du code proprement dit. Elle aborde quelques principes généraux (par exemple la gestion du code source), ainsi que la façon d'éviter certaines erreurs/problèmes courants que nous avons rencontrés.

Assurance du système :

Cette section couvre le déploiement et la gestion de la vie du code que vous avez produit. Selon le contrat spécifique, ceci peut ne pas toujours avoir la même importance, mais vous devriez assurer que l'application ou le système est capable d'opérer selon ces orientations.

Nous avons en outre inclus divers annexes qui précisent les spécifications de BT en matière de contrôle d'accès et de cryptographie. Il est important de vous assurer que votre application est conforme à ces contrôles.

1.3. Matériel supplémentaire et conseil

Selon les liens du document

2. Conception du système

Réf.	Contrôle	Motif
1	Les données doivent être classées selon la Norme de classification des données telle qu'elle est déterminée dans votre organisation. Cette norme comprend des informations sur la conception et le développement (par ex. conceptions, problèmes, exigences) ainsi que des données sur l'application. Les données doivent être conservées conformément à la politique de conservation des informations	En comprenant le type de données que votre système va contenir, vous pouvez mettre en place les protections appropriées. Il est important de sécuriser les informations de conception et de développement (en particulier les problèmes sensibles) car l'accès non autorisé peut permettre à quelqu'un de viser l'application.
2	Toutes les exigences légales et règlementaires doivent être identifiées.	Lors de la rédaction des applications qui soutiennent certains types de données (par ex. informations de la carte de crédit), il y a des exigences légales/règlementaires spécifiques qui vont affecter la façon dont vous manipulez ces données.
3	Le processus de conception doit inclure la question de savoir s'il y a des exigences de sécurité pour toutes les personnes qui travaillent sur le projet. (par ex. au R. U., seule une autorisation de sécurité peut être requise)	Les projets qui gèrent des informations marquées comme sensibles ou gouvernementales peuvent faire l'objet d'exigences quant au personnel qui sera utilisé pour travailler sur eux.
4	Utilisez les techniques d'analyse de risque pour identifier et quantifier précisément les risques de sécurité. Documentez les risques et les atténuations proposées.	Chaque application aura un ensemble différent de risques associés (par ex. dans quel environnement elle s'insère, quel niveau de données elle manipule). Il est important de vous assurer que vous les avez pris en considération et déterminé la façon de vous protéger contre ces

		risques. L'analyse des risques peut dépendre de l'environnement, plusieurs outils sont disponibles pour vous aider. OWASP a établi un guide pour identifier et évaluer les risques, qui inclut une section utile sur la façon de les classer. En alternative, Microsoft a développé un outil pour aider à identifier les menaces qui pèsent sur votre application et une autre pour comprendre votre surface d'attaque avant et après le déploiement de nouvelles applications. Pour toute question ou conseil complémentaire quant à la méthodologie, contactez le propriétaire de la norme. Remarque : La technique à utiliser peut être spécifique au projet, par exemple IS1 pour le travail agréé par HMG.
5	Définissez et documentez un calendrier d'évaluation des risques.	Ceci va assurer que les risques sont mis à jour suivant les changements de l'application ou de son environnement. De nouveaux risques peuvent être ajoutés et des mesures d'atténuation peuvent être prises.
6	Documentez l'architecture de la sécurité pour traiter tous les risques identifiés ci-dessus. Fournissez des orientations de sécurité pour des conceptions/mises en place de faible niveau des bibliothèques fiables un code secret acceptable un algorithme de hachage acceptable des longueurs minimales de clés	Ceci va assurer que vous appliquez les atténuations appropriées aux risques que vous avez identifiés et que cela est fait de façon cohérente. Cela va aussi aider à concevoir les tests d'acceptation.

	des procédures de gestion des clés la séparation du stockage/chiffrement l'accès/la transmission de références les protocoles inacceptables	
7	Définissez et documentez le cycle de vie du développement.	Disposer d'un cycle de vie du développement défini signifie qu'il existe une approche consistante et répétable pour passer des exigences/problèmes à une solution dans la production. Sans cela, il peut s'avérer difficile et long de résoudre les problèmes de sécurité. Microsoft dispose d'un modèle qui peut être utilisé comme point de départ.
8	Documentez et mettez en place les procédures suivantes : Contrôle de l'accès (définir qui peut accéder à différentes portions de l'application et comment l'accès est appliqué, conservé et révoqué) - si cet aspect n'est pas géré, il est probable que les personnes finiront pas disposer d'accès inappropriés (par ex. utilisateurs capables de réaliser des tâches administratives ou de conserver l'accès après qu'ils aient changé de travail). Synchronisation des horloges – si les horloges ne sont pas exactes, il est très difficile d'utiliser des registres de fichiers qui permettent d'identifier les problèmes, en particulier si vous devez établir une corrélation avec l'activité dans d'autres systèmes. Transferts de données – les données	L'application de sécurité est fortement dépendante de la façon dont elle est gérée pendant le cycle de vie. La définition et la mise en place des processus suggérés vous permettent d'assurer que le travail produit lors de l'étape de conception et de mise en place n'est pas négligé durant le fonctionnement de l'application. Beaucoup d'entre eux sont du sens commun et seront suivis naturellement. Il est cependant important de rendre ces processus explicites pour s'assurer qu'ils sont suivis durant la vie utile de l'application (par exemple, si l'équipe de support change).

	en transit doivent être entourées de protections appropriées à leur classification. Comment les outils et les techniques peuvent être utilisés pour attaquer le système - Ceci va vous permettre de devancer les tentatives et de mettre en place des atténuations dès le premier jour. Reprise après sinistre/repli – Ceci doit être clairement défini et testé, afin que le système puisse réagir face à des évènements imprévus. Documentez et mettez en place les procédures de développement suivantes : Inclusion des outils de test en Intégration continue – un environnement construit automatiquement permet de tester automatiquement chaque intégration et d'y détecter les problèmes courants avant que l'application ne soit distribuée. Utilisation d'outils d'analyse de la sécurité, statiques et dynamiques, comme partie du processus de développement - Ceci va aider à détecter les vulnérabilités précocement dans le processus de développement. Marquage des problèmes de sécurité dans votre répertoire des sources et détecteur de problème – en marquant les erreurs de sécurité, il est possible d'évaluer l'efficacité des atténuations/corrections. Ils peuvent aussi retourner dans l'analyse de risque du projet et les problèmes courants peuvent être partagés avec d'autres	

	développeurs pour améliorer la sensibilisation générale. Utilisation des processus de révision des pairs – La révision du code par les pairs peut identifier des problèmes qui ne peuvent pas être détectés automatiquement et prévenir les vulnérabilités qui causent des erreurs d'encodage – par exemple des erreurs logiques Déploiement d'un environnement de construction propre - l'utilisation d'un environnement de construction propre fiable réduit le risque de pollution de la sortie par un logiciel malveillant, d'anciennes bibliothèques ou des composants indésirables qui pourraient produire des défauts compromettant la sécurité. Partage des bonnes pratiques et de l'expérience – Ceci peut aider la communauté à éviter les problèmes courants qui peuvent mener à des vulnérabilités. Documentez et mettez en place les procédures de vie utile de développement suivantes : L'application opportune de correctifs de sécurité, y compris ceux destinés aux composants du tiers – pour résoudre les problèmes de sécurité qui pourraient compromettre votre application et ses données. Maintenance du matériel et du logiciel - Il est important de planifier la façon dont vous allez rester sur le matériel et le logiciel	

	supporté afin de résoudre les problèmes et de recevoir les actualisations de sécurité. Application de démarrage/arrêt – Une application d'arrêt incomplète peut mener à un comportement inattendu ou au fait que les objets ne soient pas nettoyés. Gestion de l'application et du travail - La manière dont l'application gère les travaux doit être définie, en particulier en cas de charges inhabituellement lourdes. Surveillance et alarmes – la façon dont les registres et d'autres sorties sont gérés et traités doit être définie et mise en place. Archivage/nettoyage des données – il peut être nécessaire de gérer les données de façon à les supprimer ou à les archiver quand elles ne servent plus. Ceci peut être une exigence légale pour la protection des données. Migration – Ceci va permettre de continuer le service après une mise à jour, par exemple pour résoudre les problèmes de sécurité. Contrôle des modifications – Il est important que les modifications de l'application soient contrôlées de façon à ce qu'il soit possible de comprendre si elles sont légitimes et si des problèmes surviennent qui auraient pu les causer.	
9	L'architecture de la sécurité doit être revue et actualisée à chaque cycle ou répétition. Les modifications prévues de l'application doivent	Ceci est fait pour s'assurer qu'elle reste mise à jour et intègre les modifications concernant les exigences du projet ou la mise en place.

	être revues pour s'assurer qu'elles ne compromettent pas les exigences de sécurité.	
10	Utilisez les kits d'outils standard du secteur qui sont entretenus activement plutôt que ceux que vous concevez vous-mêmes.	Les kits d'outils courants sont bien testés, sont moins susceptibles de contenir des défauts majeurs et seront appliqués rapidement en cas de détection de problèmes. Ils vous feront aussi gagner du temps ! Par exemple : OpenSSL, OpenSSH, Solaris KCF, Windows Cryptography API, Active Directory, OpenLDAP.
11	N'utilisez que les composants et les services requis pour fournir la fonctionnalité des produits. N'exposez pas les interfaces ailleurs que là où cela est requis.	En minimisant les fonctionnalités non nécessaires, vous réduisez l'exposition de votre application et pouvez diminuer l'impact d'éventuelles erreurs.
12	N'utilisez que les bibliothèques ou les composants de tiers qui sont activement entretenus et supportés – pour le logiciel supporté commercialement cela signifie que vous convenez d'un accord de support avec le vendeur durant la vie utile de l'application.	Ceci signifie que si des problèmes de sécurité sont détectés, vous pourrez actualiser les composants que votre application utilise. Pour le logiciel open-source, cela peut parfois être difficile à évaluer car la maintenance est basée sur l'engagement de la communauté. En cas de doute, vous ne devez utiliser le composant que si vous pouvez assumer sa maintenance.
13	La fonctionnalité qui renforce la sécurité doit être définie dans un ensemble minimum de lieux définis et documentés dans le code. Documentez le lieu de la fonctionnalité de renfort de la sécurité.	L'impact des changements peut être évalué plus facilement et il est plus facile de résoudre les fragilités de la sécurité. En conservant la fonctionnalité de sécurité dans un nombre limité d'endroits, il est plus facile de la comprendre et de la tester que si elle est répartie dans plusieurs lieux dans une grande base de code.

14	Assurez-vous que les versions courantes et testées des composants du tiers sont utilisées et conservées.	Des versions de logiciel plus anciennes peuvent souvent renfermer des bugs ou des fragilités qui peuvent mener à des erreurs et à un accès non autorisé à l'application.
15	L'accès au système doit se conformer à la section de contrôles Contrôle et gestion de l'accès ci-dessous.	Ceci va assurer que seules les personnes autorisées ont accès aux données de l'application.
16	Un déploiement défectueux du système résulte en un minimum de privilèges.	Ceci réduit le risque qu'un déploiement défectueux résulte en un accès hautement privilégié inutile. Si nécessaire, il doit être explicitement activé pour éviter qu'il soit oublié durant le déploiement.
17	Le logiciel ne doit fonctionner qu'avec les privilèges qu'il requiert pour réaliser sa tâche fixée. Lorsqu'un haut niveau de privilège est exigé, il doit être obtenu d'une manière bien connue comme indiqué dans l'architecture de la sécurité, puis abandonné dès que possible.	En fonctionnant avec le privilège minimum, l'impact des bugs ou des fragilités de la sécurité sera réduit car il suffira d'un accès minimum pour les exploiter.
18	Les tâches qui requièrent en permanence des privilèges élevés doivent être séparées du logiciel d'application principal et sujettes à l'examen minutieux pendant l'évaluation des pairs. Faites particulièrement attention aux tâches qui font appel/reportent à un logiciel non fiable.	Effectuer cela signifie que les erreurs ou les fragilités de la sécurité de l'application principale ne provoqueront pas un accès un privilège élevé. En réduisant la quantité de code qui fonctionne à un niveau de privilège élevé, le risque d'une fragilité de la sécurité résultant en un accès avec privilège élevé est réduit.
19	Créez un plan pour tester votre application, y compris des tests centrés sur la vérification de tous les contrôles de sécurité ou des atténuations que vous avez mises en place.	
20	Définissez des limites de sécurité qui vous permettent de contrôler : les droits d'accès aux données	Ceci assure que les contrôles sont appliqués à l'endroit correct pour protéger les données de l'application.

	l'authenticité et l'intégrité des messages la validation des entrées et des sorties.	L'échec de la mise en place de contrôles à l'endroit correct peut mener à un accès non autorisé ou à une modification des données.
21	Les systèmes doivent assurer la conservation et la préservation de l'intégrité des données.	Un contrôle faible de l'intégrité des données peut mener à une fraude, au non respect des exigences règlementaires comme Sarbanes-Oxley ou des normes du secteur, telles que PCI DSS.

3. Fourniture du système

Réf.	Politique	Motif
22	Stockez votre code source dans un environnement qui restreint l'accès de façon à ce que seules les personnes autorisées puissent faire des modifications.	Ceci va réduire le risque que soient faites des modifications non autorisées ou malveillantes qui compromettraient la sécurité de l'application. L'utilisation d'un système de contrôle de révision, tel que Git ou SVN avec accès limité à un groupe de personnes nommées est le moyen de plus facile de réaliser ce contrôle.
23	Veillez à ce que tous les changements de code sont reliés à une personne nommée. Chaque changement doit être auditable pour identifier qui a fait quoi et quand et comment il l'a fait.	Ceci signifie que tous les changements de code (par ex. qui introduit une fonctionnalité indésirable) peuvent être reliés à une personne. L'utilisation d'un système de contrôle de révision tel que Git ou SVN avec accès limité à un groupe de personnes nommées est le moyen de plus facile de réaliser ce contrôle.
24	Suivez les guides des bonnes pratiques pertinents pour votre environnement de développement ou type d'application. Assurez-vous de savoir	Ceux-ci sont souvent publiés par les vendeurs d'ensembles d'outils ou par d'autres groupes tiers. Comprendre et suivre ces bonnes pratiques vous permettra d'éviter les problèmes courants.

	quels sont les problèmes courants de la plateforme ou du langage que vous utilisez.	Nous présentons ci-dessous une sélection des guides utilisés couramment : Applications Web : les détails OWASP top 10 sur les vulnérabilités des applications web courantes. Applications Android : CERT android secure coding standard: Windows / Windows Phone : iOS / Mac OS X : Guide d'encodage sécurisé de Apple : Linux : Pour toute information complémentaire ou conseil, contactez le titulaire de la norme.
25	Utilisez un style de codage cohérent documenté.	Si les styles de codage ne sont pas cohérents avec la base de code de l'application, il peut être difficile de comprendre la logique de l'application et les problèmes ponctuels qui peuvent mener à un comportement indésirable. La plupart des vendeurs produisent un guide de style pour leur langage/environnement qu'il est recommandé d'utiliser en l'absence de toute autre exigence. Voici quelques suggestions de guides de style : Java – Norme d'encodage SEI CERT Oracle pour Java Directives d'encodage sécurisé C/C++ - CERT ou directives MISRA-C C# : Python – PEP8 php – Guide de style CodeIgniter
26	Utilisez des chaînes d'outils supportés et fiables.	Les outils supportés sont ceux qui sont rapidement disponibles et entretenus activement (soit par un seul vendeur, par ex. Microsoft Visual Studio, ou par une

		communauté, par ex. GNU Compilers Collection). Les outils fiables sont ceux qui sont fournis (et signés) par une partie fiable (par ex. Microsoft, Ubuntu) ou qui sont largement disponibles et vérifiables via des hachages bien connus (par ex. téléchargements source GCC). Si vous utilisez des outils supportés, vous recevrez des actualisations pour résoudre les bugs et d'autres problèmes de sécurité. En vérifiant la fiabilité des outils que vous utilisez, vous allez assurer que les ensembles d'outils ne vont présenter aucune fonctionnalité indésirable.
27	Entretenir les chaînes d'outils	Assurez-vous que toutes les actualisations de sécurité sont appliquées aux outils intégrés avant de les utiliser. Ceci a pour but de résoudre les problèmes avec la chaîne d'outils, qui pourrait présenter un comportement indésirable dans les binaires de sortie.
28	Utilisez les fonctionnalités du langage et de la chaîne d'outils pour vous aider à identifier ou à atténuer les problèmes simples.	En activant le compilateur ou les alertes d'exécution, vous pouvez prévenir les problèmes qui peuvent mener à des fragilités de sécurité ou à des bugs (par ex. une protection de pile forte pour gcc avertit d'éventuelles situations de débordement potentiel de tampon ; ou /GS ou /SafeSEH pour VisualC++ pour rendre les débordements de tampon basé sur la pile plus difficiles à exploiter).
29	Utilisez des fonctionnalités de système opératif qui atténuent les vecteurs d'attaque courants.	Ceci est spécifique au système opératif mais vous devez activer des fonctionnalités de sécurité qui sont disponibles pour accroître la protection de votre application, par exemple ASLR ou DEP. Un support Microsoft basé sur des systèmes opératifs est disponible.

		Pour linux, ASLR et DEP sont normalement activés par défaut (pour des kernels postérieurs à 2.6.12). SELinux est un moyen de renforcer les politiques de contrôle d'accès et peut être utilisé pour réduire l'impact sur une exploitation. Redhat dispose d'un bon guide sur les politiques de développement.
30	Désinfectez toutes les entrées conformément à l'architecture de la sécurité avant le traitement. Convertissez les entrées en une forme canonique avant la désinfection.	Si une entrée-utilisateur est utilisée comme base de commandes ou pour les opérations de la base de données, l'entrée peut être utilisée pour provoquer une opération indésirable, par ex. injection SQL. Les entrées peuvent souvent prendre des formes inhabituelles, par exemple des objets sérialisés dans java et il est donc important d'envisager tous les cas où un utilisateur/intermédiaire pourrait modifier ou injecter des contenus. Les entrées qui ne sont pas converties peuvent contourner la désinfection en utilisant des ensembles de caractères que la désinfection n'a pas été conçue pour manipuler.
31	Désinfectez toutes les sorties vers d'autres systèmes conformément à l'architecture de la sécurité.	Lorsqu'une entrée-utilisateur est utilisée comme sortie (soit vers une autre application, soit de retour vers l'utilisateur), l'entrée peut être manipulée pour causer un comportement indésirable dans ce qui reçoit la sortie (par ex. script inter sites)
32	Mettez la cryptographie en place conformément aux contrôles dans la section cryptographie ci-dessous.	La cryptographie peut être complexe et difficile à obtenir. En suivant les contrôles appropriés, vous pouvez vous assurer que le niveau de cryptographie que vous utilisez est approprié aux informations que vous souhaitez protéger
33	Mettez en place des mécanismes pour éviter des opérations de mémoire dangereuses. Voir OWASP	Certaines opérations de mémoire peuvent faire qu'un utilisateur sans privilège puisse manipuler les contenus de la mémoire. Dans certains contextes, ceci peut mener à la manipulation du flux de programme (par ex.

		contrôles de contournement) ou l'exécution d'un code arbitraire.
34	Utilisez des outils anti-altération là où l'indique l'évaluation des risques	Le fait de permettre la modification du code d'un firmware ou d'une application peut occasionner le contournement des fonctions de sécurité. Ceci peut s'appliquer à des dispositifs mobiles où vous pouvez vouloir arrêter votre application de fonctionner sur des dispositifs « enracinés » ou « jailbreakés ». Vous pouvez souhaiter de faire cela pour qu'il soit plus difficile pour les utilisateurs de faire une analyse/débogage de l'exécution pour inverser la fonctionnalité de l'application de l'ingénieur.
35	Utilisez des techniques anti-inversion là où l'indique l'évaluation des risques	L'ingénierie d'inversion peut révéler des fonctions de sécurité. Cependant, il est essentiel de vous souvenir que la plupart des techniques anti-inversion peuvent seulement augmenter le temps nécessaire à l'ingénieur d'inversion, plutôt que de l'empêcher complètement. Elles ne doivent donc pas être la seule forme de défense de la sécurité.
36	Évitez les fichiers temporaires. N'utilisez pas tmpnam() ou similaire pour générer des noms de fichiers.	L'utilisation de fichiers temporaires peut provoquer une répartition des données dépassant largement un disque qui peut être recouvrable si l'accès non autorisé est obtenu dans le système. tmpnam() doit être évité car il se peut qu'entre la génération du nom et l'ouverture du fichier, un autre processus ait créé un fichier avec le même nom en utilisant tmpnam. Ceci peut faire que l'autre processus soit capable d'affecter l'intégrité des données stockées ou de lire les données auxquelles il ne devrait pas avoir accès.
37	N'utilisez pas de mots de passe prédéfinis (codés en dur).	Les mots de passe qui sont prédéfinis dans l'application vont normalement être connus et partagés entre les utilisateurs. Ceci peut permettre l'accès d'utilisateurs non

		autorisés ou il peut être difficile de prouver qui a eu accès, car ils ne sont pas assignés à un seul utilisateur. Il n'est pas non plus possible d'actualiser et de gérer un mot de passe codé en dur conformément aux spécifications de BT en matière de gestion de compte.
38	Mettez en place des mécanismes pour éviter l'accès non autorisé à des données de la mémoire.	En fonction de votre plateforme et de l'évaluation des risques, vous pouvez avoir besoin de prendre des mesures pour empêcher quelqu'un qui a accédé à votre équipement et qui exécute votre application de récupérer des données sensibles à partir de votre mémoire. Ceci peut consister en : • remplacer des portions de mémoire quand elles ne sont plus utiles • verrouiller la mémoire • Utilisez des constructions de cadre conçues pour protéger les informations, par ex. le SecureString classe dans .Net
39	Les rapports d'erreurs peuvent contenir suffisamment d'informations pour identifier la cause des problèmes. Lorsque les erreurs sont affichées pour les utilisateurs, elles ne doivent cependant pas révéler d'informations sur l'intérieur des applications. Par exemple, n'utilisez pas les types d'exception généraux pour fournir les situations d'erreur spécifiques.	Des rapports d'erreur incomplets peuvent freiner les enquêtes sur les problèmes et occulter les activités non autorisées. Une exploitation va souvent générer des erreurs au premier stade pendant l'obtention de l'accès et si elles sont correctement enregistrées, elles peuvent faciliter l'enquête sur le problème. Il est difficile de comprendre ce qui a causé la situation d'erreur si seule une exception générale est enregistrée. Si la situation d'erreur a été causée par une attaque, il est plus difficile de découvrir quelles opérations ont été entreprises.
40	Tout le logiciel doit être testé avant la migration dans l'environnement réel.	Le logiciel non testé peut : Présenter les risques de sécurité inacceptables Ne pas fonctionner selon les exigences, en

	Adoptez le plan de test que vous avez créé comme partie de la conception de l'application.	particulier les exigences de sécurité détaillées dans l'Enregistrement de la sécurité Affecter les opérations existantes de façon adverse.
41	Tous les contrôles de sécurité doivent être testés spécifiquement pour garantir qu'ils ne peuvent pas être contournés.	Les vulnérabilités non identifiées des contrôles de sécurité doivent être exploitées en environnement réel.
42	Les données des tests doivent être effacées après le délai déterminé par le propriétaire des données.	La diffusion des données de test pourrait donner un aperçu du système du donateur.

4. Assurance du système

Réf.	Contrôle	Motif
43	Les vulnérabilités de la sécurité identifiées dans le code ou tout composant utilisé par le code seront catégorisés selon le critère d'évaluation de la vulnérabilité de BT. Les vulnérabilités doivent être résolues conformément à l'échelle temporelle associée à chaque catégorie. Ceci peut être réglé à partir de la position de risque de la plateforme dans laquelle se trouve l'application.	Les vulnérabilités non résolues peuvent être utilisées comme partie d'une attaque à l'application ou au système.
44	Tracez et marquez les changements pour résoudre les problèmes de sécurité explicitement dans le contrôle des	En traçant les résolutions des problèmes de sécurité, vous pourrez réaliser rapidement l'audit de votre application/environnement et prouver que ces

	changements.	changements ont été appliqués.
45	Installez les versions d'application qui ont été construites à partir de la source à l'aide de l'intégration continue.	Le déploiement à partir d'un environnement propre construit signifie que l'application commence sa vie opérationnelle avec un nouvel ensemble de code et que les problèmes peuvent être suivis depuis le code source qui les a présentés.
46	Assurez-vous que la dernière version de l'application et tous les composants du tiers sur laquelle elle s'appuie sont utilisés.	En actualisant les composants du tiers, le risque qu'un problème de sécurité dans un composant soit exploité est réduit.
47	Appliquez des correctifs à l'environnement qui héberge l'application conformément à la politique de correction.	Il est important d'appliquer des correctifs de sécurité régulièrement car les versions de logiciel plus anciennes deviennent souvent des cibles d'exploitation. Un problème de sécurité dans l'environnement d'hébergement peut faire que l'application soit en risque.
48	Les domaines de processus suivants doivent être définis et documentés : Application de correctifs de sécurité Contrôle de l'accès Démarrage/Arrêt Synchronisation des horloges, Gestion de l'application et du travail Transferts de données Surveillance et alarmes Gestion des problèmes et des escalades Sauvegarde et Récupération Archivage Migration Contrôle des changements Reprise après sinistre/repli	Une défaillance de la documentation des procédures d'opération signifie souvent que d'importants processus ont été négligés ou que l'équipe manque de connaissances sur la façon de les réaliser.

	Entretien du matériel et du logiciel Enregistrement, évaluation et mesures en cas d'évènements inhabituels.	

5. Annexes

5.1. Contrôle de l'accès

Réf.	Politique	Motif
49	Définissez les droits d'accès au système pour les utilisateurs et les autres systèmes qui interagissent avec lui. Les droits d'accès doivent être basés sur : les opérations que le système doit réaliser l'interaction système vers système l'accès de l'utilisateur aux informations et à la politique des systèmes les utilisateurs doivent disposer des privilèges minima pour accomplir leur travail.	une faible définition des fonctions peut occasionner la réalisation d'opérations accidentelles ou malveillantes dans le système.

50	Les comptes à privilèges partagés* doivent être gérés comme défini.	Sans contrôle formel il est difficile de suivre ceux qui sont responsables des changements qui ont un impact sur la sécurité.
51	Les processus de conception de l'application doivent fonctionner avec les droits d'accès minima requis pour une opération correcte. Les opérations que le système doit réaliser. L'interaction système vers système. N'utilisez pas de comptes avec des privilèges pour les processus de l'application. Documentez tous les accès avec privilège. L'escalade des privilèges ne doit être connue que par les APIs. L'escalade des privilèges ne doit être faite que pour la durée minimale requise.	Les processus d'application qui fonctionnent avec un excès de privilèges peuvent être exploités pour obtenir l'accès aux privilèges de données ou de système.
52	Les sessions d'utilisateur doivent être interrompues après un maximum de 30 minutes d'inactivité. Lorsqu'un arrêt se produit, l'écran doit être nettoyé de toutes les informations affichées.	Les arrêts limitent la possibilité d'accès non autorisé si le système est laissé sans surveillance.
53	Une session d'utilisateur ne doit pas excéder 12 heures.	Pour être sûr que les utilisateurs s'enregistrent et se ré-authentifient au moins toutes les 12 heures.

54	Les contrôles d'accès basés sur la fonction et le privilège doivent être fournis sur tous les ports de gestion, qu'ils soient locaux (artisan/ terminal de console ou serveur terminal) ou à distance (via EMS). Les ACL doivent être appliqués sur toutes les interfaces de gestion (IP), en restreignant l'adresse source et le port et avec protocole invoqué ; en particulier la capacité de restreindre l'accès à ICMP, HTTP, SNMP, FTP, TFTP, SSH, Telnet et des protocoles de routage avancés tels que OSPF.	Les interfaces de gestion exposées peuvent être exploitées en vue d'un accès non autorisé.
55	Seuls des protocoles de gestion authentifiés en toute sécurité doivent être utilisés. SNMP v3 (AuthnNoPriv au minimum) SSHv2 - voir section de la cryptographie HTTPS - voir section de la cryptographie.	Les protocoles de gestion exposés peuvent être exploités en vue d'un accès non autorisé.

5.2 Cryptographie

Réf.	Politique	Motif
56	Les bibliothèques cryptographiques actuelles doivent être utilisées.	Les bibliothèques cryptographiques sont actualisées régulièrement. En plus de l'actualisation des paquets de logiciels en ligne avec les indications du vendeur, les

		paquets de cryptographie doivent être revus et actualisés régulièrement.
57	Pour le chiffrement, n'utilisez que des suites de chiffrement standard approuvées par le secteur. Par ex. pour TLS SSLv2. Prenez note de l'avertissement ci-dessus. En cas de doute, demandez conseil à ITSAC.	Les chiffres non approuvés peuvent introduire des vulnérabilités.
58	La dernière version de TLS doit être utilisée pour de nouveaux déploiements. Les V1, 2 et 3 SSL ne doivent pas être utilisées.	Les versions précédentes allant jusqu'à TLS1.0 incluse ne sont plus considérées comme fiables.
59	La confidentialité de transmission parfaite doit être activée.	Les algorithmes de la confidentialité de transmission parfaite évitent que les messages capturés soient décryptés même si la clé privée d'authentification est compromise pour l'avenir.
60	Il ne faut pas utiliser de certificats auto-signés	Les certificats auto-signés nuisent au bénéfice de l'authentification du point final et diminuent également de façon significative la capacité d'une personne à détecter une attaque de l'homme du milieu.
61	Les mots de passe doivent être protégés en utilisant une fonction mathématique à sens unique non réversible (par ex. l'algorithme de hachage) avec un facteur unique aléatoire (Salt) par mot de passe.	Les fichiers de mot de passe stockés peuvent être extraits et par conséquent toutes les entrées doivent être protégées pour éviter la récupération des mots de passe en texte clair.
62	Les mots de passe protégés doivent être stockés loin des fichiers de configuration	Il ne doit jamais être possible de récupérer les mots de passe protégés par un répertoire croisé, un snmpwalk,

	des systèmes et disposer d'un contrôle d'accès de façon à ce que seuls les utilisateurs privilégiés appropriés puissent lire ou copier leur contenu.	un script de configuration ou un autre mécanisme qui pourrait permettre des tentatives de craquage hors ligne.

5.2.1 Contrôles de cryptographie web

Réf.	Politique	Motif
63	Les cookies qui stockent ou qui sont utilisés pour l'accès Confidentiel ou à un niveau de données supérieur doivent être transportés en toute sécurité.	Les cookies peuvent être volés au moyen de diverses méthodes telles que XSS et reniflement.
64	Les contenus sécurisés et non sécurisés ne doivent pas être mélangés sur la même page.	Les contenus non sécurisés peuvent éventuellement voler des informations sécurisées du contenu sécurisé.
65	Il faut utiliser TLS pour toutes les pages d'enregistrement et toutes les pages authentifiées. L'accès à la page d'enregistrement et à toutes les pages authentifiées suivantes doit se faire exclusivement par TLS. L'accès à la page d'enregistrement et à toutes les pages authentifiées suivantes doit se faire exclusivement par TLS.	La non utilisation de TLS pour la page de destination de connexion permet à un attaquant de modifier l'action du formulaire de connexion, ce qui fera que les références de l'utilisateur seront postées dans un lieu arbitraire. La non utilisation de TLS pour les pages authentifiées après la connexion permet à un attaquant de visualiser la session ID encodées et compromet la session authentifiée de l'utilisateur. La non utilisation de TLS peut résulter en une attaque de l'homme du milieu.

66	Les clients doivent être avertis de ne pas mettre en cache les données sensibles.	Le protocole TLS fournit uniquement la confidentialité pour les données en transit mais ne permet pas de résoudre les problèmes d'éventuelles pertes de données des clients.
67	Utilisez les normes de signature numérique nationalement et internationalement acceptées.	L'utilisation de techniques de signatures numériques non normalisées peut résulter en une confiance déplacée dans la signature ; la régulation nationale et internationale permet de contrôler les résumés numériques et d'effectuer des contrôles cryptographiques de l'importation, exportation et domestique dans le monde entier.
68	Les certificats de caractère générique ne doivent être utilisés en aucune circonstance.	Les certificats de caractère générique sont la cible la plus attractive. Ils peuvent certifier un serveur non prévu et rendre un domaine de chiffrement vulnérable. Si un serveur ou sous domaine est compromis, tous les sous domaines seront compromis. Si le certificat de caractère générique doit être révoqué, tous les sous domaines vont avoir besoin d'un nouveau certificat.

5.2.2 Gestion générale des clés

Réf.	Politique	Motif
69	Les clés cryptographiques pour tous les nouveaux déploiements doivent au moins être celles prévues par la norme ou dépasser les normes minimales.	Les clés courtes ou faibles peuvent facilement être compromises durant la vie utile des données.

70	Il faut générer des clés symétriques et asymétriques à l'aide d'outils et de bibliothèques approuvés	Les outils et les bibliothèques non approuvés peuvent générer des clés faibles.
71	Les mots de passe qui contrôlent l'utilisation de la cryptographie doivent fournir une protection équivalente à la clé proprement dite.	Les mots de passe faibles annulent la force de la clé cryptographique.
72	Un gestionnaire sénior de BT doit être responsable de la sécurité du matériel des clés.	Il aura l'ancienneté, la capacité et la fiabilité qui correspondent à la sécurité des données que les clés vont protéger.
73	Le gestionnaire sénior des clés doit s'assurer que les responsabilités détaillées dans la colonne sont attribuées et réalisées par du personnel dûment qualifié et formé :	Responsabilités Politique de la gestion des clés Production ou acquisition de clés Conception de la distribution des clés, durées des clés, révocation et structure de compte Création d'une procédure de gestion des clés Fonctionnement de la gestion des clés Protection des clés privées et des matériaux associés Procédures d'urgence telles que révocation Audit des opérations des clés Récupération des clés Destruction des clés

Contrôle du document

Normes d'orientation du secteur pour les tiers en matière de « Encodage sécurisé »

Auteur : Sécurité de BT

Émission 1, publiée en octobre 2016