Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

PREMIER MINISTRE

Secrétariat général
de la défense nationale

Direction centrale de la sécurité des systèmes d'information

Paris, le 23 octobre 2008

N° 2411/SGDN/DCSSI/SDR
Référence : VUL/I/01.1

INSTRUCTION

INTERPRETATION DU VLA.4/VAN.5 DANS LE DOMAINE DU LOGICIEL

Objet : Interprétation du VLA.4/VAN.5 dans le domaine du logiciel

Application : A compter de la parution

Diffusion : Publique

Vérifié par le responsable qualité	Validé par le chef du centre de certification
[ORIGINAL SIGNE]	[ORIGINAL SIGNE]

Suivi des modifications

Révision	Date	Modifications
1	23/10/08	Création

TABLE DES MATIERES

1. OBJET DE L'INSTRUCTION 4

2. REFERENCES 4

3. PROBLEMATIQUE 4

4. ILLUSTRATIONS 5

5. ACCEPTATION DES DEMANDES DE CERTIFICATION VLA4 6

1. Objet de l'instruction

Cette instruction précise la position du schéma français vis-à-vis des composants d'assurance AVA_VLA.4 et AVA_VAN.5.

2. Références

- [CC v2.3] : Critères Communs Parties 1-2-3 et CEM ; version 2.3 ; août 2005 ; réf. : CCMB-2005-08-001 à 004
- [CC v3.1] : Critères Communs Parties 1-2-3 et CEM ; version 3.1 ; septembre 2007 ; réf. : CCMB-2007-09-001 à 004

3. Problématique

Si la communauté des évaluateurs, utilisateurs, développeurs et certificateurs a une bonne compréhension de la signification du AVA_VLA.4 / AVA_VAN.5 (par la suite, VLA4 et par extension, « niveau VLA4 ») dans le domaine de la carte à puce et des microcontrôleurs sécurisés, on constate qu'il n'en est pas de même pour le domaine logiciel à la date de rédaction de ce document. Pour tenter d'en comprendre les raisons, il faut rappeler quelques-unes des particularités des évaluations de cartes et les mettre en regard de ce qui constitue l'état actuel des évaluations logicielles :

1) Le nombre d'évaluations de cartes et composants dans le cadre des schémas français et allemand est très important et représente probablement la catégorie de produits pour laquelle il y a le plus fort retour d'expérience.

   Le nombre d'évaluations de logiciels se répartit sur de nombreuses catégories (pare-feux, systèmes d'exploitation, signature, chiffrement, etc.). Elles sont réalisées au sein de nombreux schémas d'évaluation ce qui fait qu'il est plus difficile de consolider un retour d'expérience.

2) Les évaluations de cartes et de composants se font essentiellement au niveau VLA4.

   Les évaluations logicielles sont le plus souvent faites à des niveaux AVA_VLA2 ou 3, sauf lorsqu'il existe des hypothèses simplificatrices (cf. 5).

3) Les évaluations de cartes et de composants se font par rapport à des profils de protection peu nombreux ce qui facilite les comparaisons.

   En revanche, pour des mêmes types de produits logiciels, il peut exister de nombreux profils de protection ce qui exclut toute possibilité de comparaison.

4) Les utilisateurs de cartes évaluées et certifiées sont très concernés par la façon dont se déroulent les évaluations (en particulier, par la compétence des centres d'évaluation) et par le résultat des analyses de vulnérabilités.

   Les évaluations de produits logiciels sont essentiellement commanditées dans le but de disposer d'un avantage concurrentiel. Il est rare que les utilisateurs finaux soient impliqués dans ce processus. Les développeurs (et les schémas de certification) n'ont donc pas d'intérêts particuliers à pousser les centres d'évaluation vers le haut en termes de compétence sur les attaques.

5) Lors des évaluations de cartes, il n'y a généralement pas d'hypothèses simplificatrices ayant pour conséquence l'élimination de certains chemins d'attaque.

   Dans le cas des évaluations de logiciels, il y a généralement de nombreuses hypothèses simplificatrices (hypothèses sur l'environnement technique et organisationnel) qui limitent la portée des attaques pouvant être réellement menées sur le produit. Autrement dit, la résistance d'un produit à des attaques de niveau VLA4 peut être systématiquement atteinte du fait qu'en pratique, les hypothèses éliminent les possibilités d'attaques.

6) La communauté impliquée dans les évaluations de cartes et de composants est très active dans la spécification des méthodes d'évaluation des cartes, y compris sur l'aspect lié aux attaques. En pratique, le travail de cette communauté permet de définir l'état de l'art des attaques de haut-niveau.

Il est difficile d'identifier une telle communauté dans le domaine du logiciel.

7) La notion « d'erreur de construction », pouvant entraîner des vulnérabilités, est mal acceptée dans le monde de la carte (même s'il en existe).

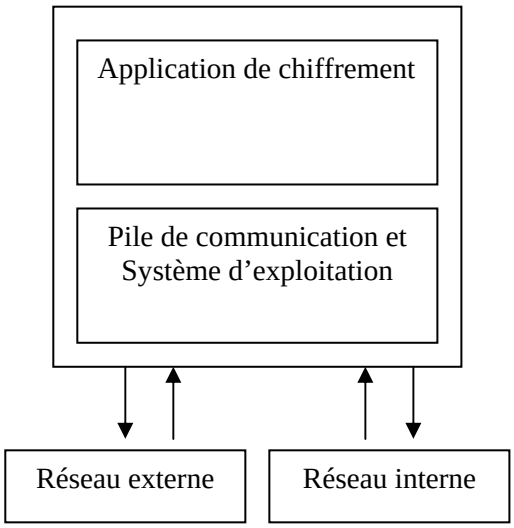
   Par la force des choses, les utilisateurs de logiciels en sont venus à considérer comme une fatalité l'existence d'erreurs de construction (bogues) entraînant la création de vulnérabilités qui représentent l'essentiel de celles qui sont découvertes. Un effet pervers de ce phénomène est que, même si les fonctions de sécurité d'un produit sont susceptibles de résister à un niveau élevé d'attaque, la confiance dans ce résultat est en pratique assez faible car des erreurs de construction non détectées durant l'évaluation sont susceptibles de remettre en cause l'efficacité de ces fonctions, alors même qu'elles ne sont pas forcément impliquées dans ces erreurs[1].

De ces constats, il apparaît que la signification du VLA4 dans le domaine logiciel peut être très variable (ce constat est d'ailleurs également valable pour les autres niveaux de AVA_VLA). Pratiquement, deux produits fonctionnellement identiques peuvent être certifiés avec le même niveau AVA_VLA alors que dans un cas, la sécurité du produit reposera sur des hypothèses d'environnements techniques et organisationnels très fortes tandis que ce ne sera pas le cas pour l'autre.

Cette souplesse autorisée par les CC (dans une certaine mesure, ils sont conçus pour) peut créer des situations que l'on pourrait qualifier de trompeuses pour l'utilisateur final, pas forcément au fait de ces subtilités. Le schéma français peut contribuer à éviter que ce type de situation se perpétue en proposant des règles pour l'acceptation des dossiers de demande de certification. Il faut néanmoins être conscient de la limite de ces règles, les accords de reconnaissance actuels (CCRA, SOG-IS) obligeant la DCSSI à formellement reconnaître les certificats des pays émetteurs qui ne les appliqueraient pas.

4. Illustrations

Considérons un équipement de chiffrement de réseau IP. Une architecture possible serait la suivante :

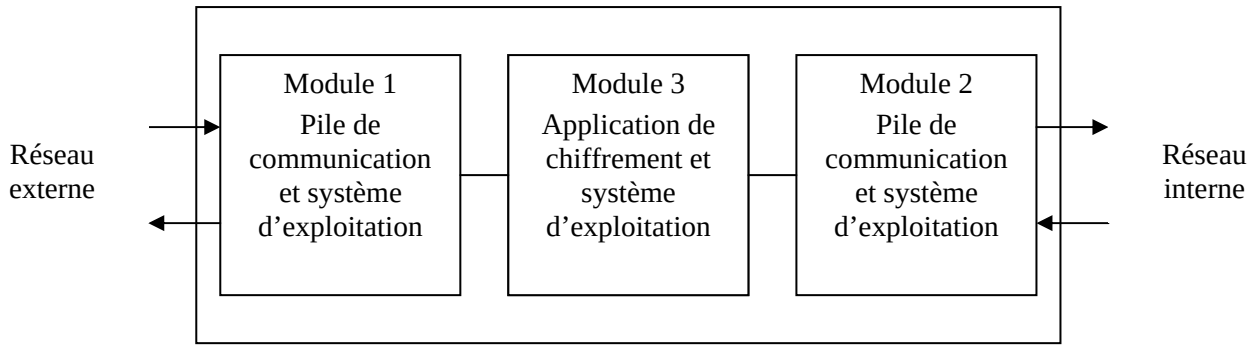


Architecture 1

Dans cette architecture, le processus de chiffrement partage une même pile protocolaire pour ses communications entrantes et sortantes, l'ensemble s'exécutant sur un système d'exploitation commun.

[1] Il peut sembler surprenant que l'évaluation ne soit pas en mesure d'identifier ces anomalies. Il faut rappeler que sur un logiciel de grande taille et très répandu, typiquement un système d'exploitation, il existe de nombreux experts qui passent un temps important à étudier et à travailler sur chacun de ces sous-systèmes. Leur puissance collective d'analyse est nettement supérieure à celle que peut mettre en œuvre un centre d'évaluation, faisant travailler un nombre limité de personnes en temps contraint pour trouver des vulnérabilités.

Dans l'architecture suivante, l'environnement des moyens de communication IP et de chiffrement s'exécute sur des plates-formes physiquement séparées, ces plates-formes étant reliées par un bus de communication matériel dédié.



Architecture 2

Sans entrer dans le détail des hypothèses de conception, il est probable que, même à hypothèses identiques (hypothèse de confiance « a priori » sur le système d'exploitation et la pile de communication par exemple), l'architecture 2 inspirera une confiance plus grande que la première. Une raison est qu'il est intuitivement plus difficile de compromettre les modules 2 et 3 à partir d'une vulnérabilité exploitée sur le module 1 que dans le cas de l'architecture 1 où une vulnérabilité sur n'importe quelle composante (système d'exploitation, pile de communication, application) est susceptible de compromettre toutes les autres composantes. Cette confiance plus importante est déterminée par le concept de défense en profondeur, élément qu'il est difficile de prendre en compte dans une évaluation CC.

Cet exemple est une illustration des différences de signification possibles pour un composant AVA_VLA.x et particulièrement, VLA4. D'autres cas peuvent être évoqués :

- Un produit complexe inspire en général moins confiance qu'un produit simple. Une illustration peut être tirée du monde de la carte. Il existe des cartes dont les applications s'appuient sur un interpréteur Javacard et d'autres dont les applications sont programmées en natif. Un interpréteur Javacard propose des fonctionnalités à un niveau d'abstraction plus élevé que le processeur ce qui entraîne une plus grande complexité du code et des tests que pour une carte programmée en natif. Inversement, si l'on ne considère que le point de vue de l'application, sa programmation sera simplifiée si elle s'appuie sur une Javacard puisqu'elle utilise des primitives plus riches que celles proposées par le processeur. On voit par cet exemple que le critère de complexité doit être interprété au cas par cas.
- Beaucoup d'applications logicielles s'appuient sur d'autres produits pour s'exécuter et, en particulier, sur un système d'exploitation qui peut être généraliste (Microsoft® Windows®, Linux). En quantité de code exécutable, les applications en question peuvent ne représenter qu'une part infime du code effectivement présent (et exécuté) par le produit complet (application, système d'exploitation, matériel et logiciels embarqués). La pratique montrant que la probabilité d'avoir un bogue susceptible d'entrainer une vulnérabilité exploitable sur les produits sous-jacents est importante, le risque est grand que les fonctions de sécurité implantées dans les applications en question puissent devenir inopérantes.

5. Acceptation des demandes de certification VLA4

La problématique exposée dans les paragraphes précédents n'est pas nouvelle (elle est aussi ancienne que les critères d'évaluation). Pour autant, elle n'a pas forcément donné lieu à une doctrine bien établie concernant ce qu'il est raisonnable d'accepter en matière d'évaluation au niveau VLA4. La raison est que, jusqu'à présent, il n'a pas été exposé de règles applicables de façon systématique et objective. Ce document n'a pas

pour objectif de fournir de telles règles mais, néanmoins, le schéma français a souhaité évoquer quelques principes directeurs sur ce qu'il semble raisonnable d'accepter ou de refuser au titre d'une demande de certification au niveau VLA4.

Principe 0 : la DCSSI se donne le droit de refuser une demande de certification visant le niveau VLA4 qu'elle estimerait trompeuse.
Principe 1 : l'architecture du produit doit contribuer à la sécurité en faisant en sorte que les fonctions de sécurité du produit ne puissent être mises en défaut que si au moins deux vulnérabilités distinctes sont exploitées. *Exemple : voir illustrations dans le présent document, partie 4.*
Principe 2 : les hypothèses sur l'environnement technique doivent être réalistes et en tout ou partie vérifiables.
Principe 3 : les hypothèses sur l'environnement technique (matériel et logiciel) doivent être détaillées. Le développeur doit fournir à l'évaluateur la preuve du réalisme de ces hypothèses. *Exemples : l'hypothèse « système d'exploitation de confiance » doit être détaillée. La question de la protection mémoire doit être développée : selon le système d'exploitation utilisé, comment le programme évalué gère-t-il le problème du swap, du coredump, d'un éventuel « debugger » qui serait en train de le tracer, des copies mémoire d'éléments secrets que certaines fonctions de librairies pourraient réaliser (que se passe-t-il si on passe une clé de chiffrement à une fonction C non écrite par le développeur ? Risque-t-elle d'être copiée en mémoire et de perdurer au-delà du moment où l'original de la clé est surchargé par des « 0 » au sein du programme principal ?).* *L'hypothèse «le matériel de confiance » doit également être détaillée. On pourra notamment proposer des hypothèses du type «le processeur est exploité dans son mode nominal de fonctionnement. Dans ce mode, le processeur gère au minimum deux niveaux de privilège (un niveau utilisateur et un niveau superviseur) » et « toute transition vers un mode du processeur autre que son mode nominal est impossible ».* *Dans le même ordre d'idée, il existe des publications montrant l'exploitation de vulnérabilités touchant en particulier l'implémentation de la cryptographie en exploitant les propriétés du cache mémoire, les mécanismes de prédiction de branchement, la rémanence des mémoires et les traitements associés au calcul des sous-clés.* *Voir également la note associée au principe 4.*
Principe 4 : les hypothèses d'environnement organisationnel doivent être réalistes pour le type de produit concerné. *Note : il est acceptable de définir plusieurs niveaux de résistance aux attaques selon les fonctions de sécurité proposées par le produit plutôt que de clamer que toutes les fonctions de sécurité visent le niveau VLA4 et introduire pour ce faire des hypothèses d'environnement très simplificatrices (du point de vue de l'analyse de vulnérabilité). La cible de sécurité sera alors multi-niveau pour AVA_VLA.*
Principe 5 : un produit dont la résistance aux attaques de fonction de sécurité VLA4 ne repose que sur des hypothèses d'environnement n'est pas acceptable.
Principe 6 : une mesure (métrique à définir) de la complexité du logiciel évalué devrait être fournie et vérifiée par le laboratoire.
Principe 7 : l'existence de compétences couvrant les fonctionnalités de sécurité du produit chez le développeur devrait être validée.
Principe 8 : avant toute demande de certification au niveau VLA4, le commanditaire devrait valider avec la DCSSI l'acceptabilité de sa cible de sécurité.