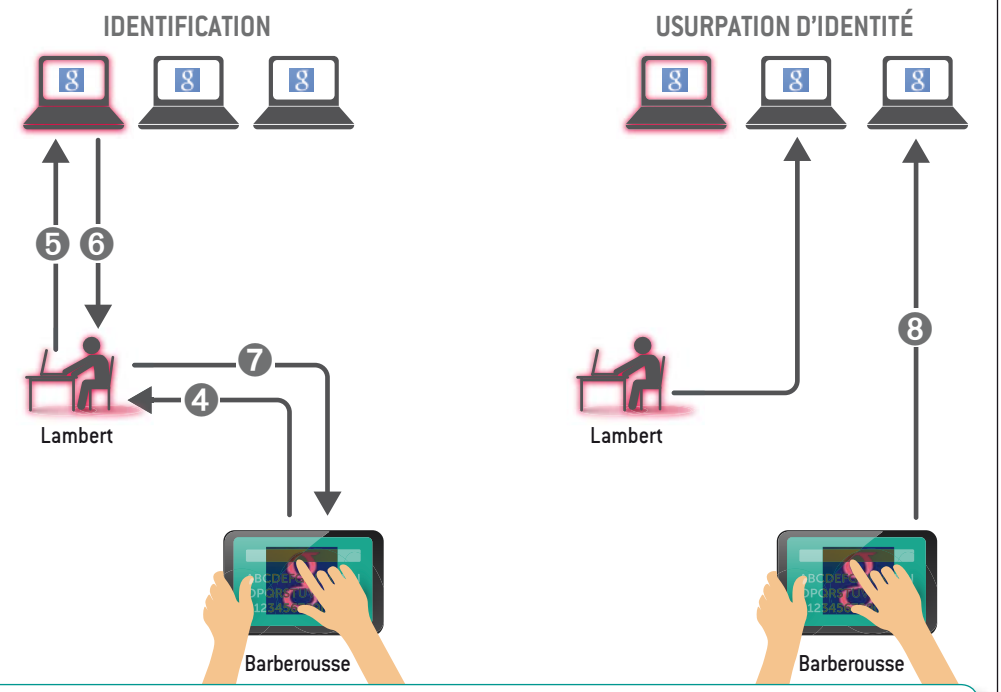
ET D'ACCÈS AUX APPLICATIONS *GOOGLE*



bert. Celle-ci répond par une demande d'identification ❸, que Barberousse répercute à Lambert ❹. Ce dernier s'identifie ❺ auprès du serveur dédié à l'authentification des utilisateurs de *Google*. Ce serveur renvoie ❻ la preuve de l'identité de Lambert à son ordinateur, lequel, conditionné par l'application malveillante de Barberousse, lui communique ladite preuve ❼. Barberousse se sert de cette preuve pour se connecter ❽ par le compte de Lambert à une application *Google*, par exemple l'agenda ou la messagerie de Lambert ; dès lors qu'il s'est connecté avec l'identifiant et le mot de passe de Lambert, ce dernier est aussi simultanément connecté sur l'application sportive, qu'il consulte alors, toujours de bonne humeur, pendant que Barberousse lit son emploi du temps, vole ses secrets ou envoie des messages depuis son ordinateur...

d'erreurs dans la logique présidant à la conception du système.

La première erreur logique possible consiste à envoyer la clef sans la cacher, en d'autres termes à transmettre la clef en clair, ou d'une façon qui permet de la voler. Le vol d'identité résultant d'informations de connexion d'un utilisateur à un service en ligne (une banque en ligne, par exemple) non chiffrées sur Internet fournit un exemple trivial d'une telle erreur logique. Dans ce cas simple, un attaquant n'a qu'à intercepter la communication entre l'usager et son fournisseur de service pour dérober les informations confidentielles permettant de se connecter à la place de l'usager. De telles failles se sont révélées récemment sur des services bien connus du grand public. On peut citer une version ancienne du protocole de connexion des particuliers aux services de *Google* (dit *Google Single Sign-on*), qui a la particularité d'étendre à toutes les applications de *Google* la connexion d'un utilisateur à une seule application.

Ce genre d'attaque permet, par exemple, d'accéder à l'agenda d'une personne après qu'elle a entré son nom d'utilisateur et son mot de passe. Elle repose sur la ruse consistant à pousser l'utilisateur à se connecter à une application malveillante conçue pour l'intéresser, mais secrètement contrôlée par l'attaquant. Il pourra s'agir, par exemple, d'une application affichant les derniers résultats sportifs. L'attaquant intercepte le message envoyé par l'utilisateur pour accéder à cette application. Cette interception est intéressante pour l'attaquant, car ce message a le défaut de ne pas être spécifique à une application donnée, de sorte que l'attaquant peut réutiliser les informations de connexion pour se connecter à d'autres applications depuis le compte *Google* de l'utilisateur, par exemple à sa messagerie, à l'agenda, au service cartographique, à ses archives personnelles, etc. *(voir l'encadré ci-contre)*.

Ce cas illustre qu'en matière de sécurité informatique aussi, le diable est dans les détails, dans des détails de conception infimes, parfois presque invisibles, mais qui peuvent avoir d'importantes répercussions sur la sécurité. Ces failles dans la conception d'un système de protection des données sont d'autant plus graves que la faiblesse qu'elles engendrent affecte des services déjà déployés, et que le déploiement d'un patch ne suffit pas à les réparer. Dans le cas du protocole de connexion aux services de *Google*, changer un message de connexion utilisé par toutes les applications n'est guère réaliste, car cela a des effets à la fois sur la pratique de l'utilisateur et sur l'accessibilité des autres applications. Il est en effet souvent important de maintenir la compatibilité entre les divers protocoles donnant accès à un faisceau d'applications, même si l'un d'eux est défectueux. Une révision complète du système de connexion des services de *Google* a donc été nécessaire pour supprimer la faille.

Que faire pour éviter ce genre de difficulté ? Pour l'essentiel, il s'agit de tester de façon complète, avant de déployer le système de protection, la sûreté des procédures utilisées pour faire circuler les coffres-forts (les informations cryptées) qui protègent les données. Différents modèles théoriques de mise en œuvre existent. Dans le modèle dit symbolique, on procède en supposant d'abord que les primitives cryptographiques choisies sont incassables, puis en représentant par des symboles toutes les opérations effectuées par le système de sécurité. Sur la base d'une définition rigoureuse de la sécurité à obtenir, une technique mathématique rigoureuse permet ensuite de s'assurer que le protocole résiste à toutes les formes d'attaques décrites au sein du modèle utilisé. Une fois cette étape franchie, on peut passer à la mise en œuvre du système de sécurité.

... ou dans l'implémentation

Nous avons maintenant soigneusement choisi des coffres-forts adéquats (des algorithmes de cryptage), organisé scrupuleusement leur transport en nous méfiant de toutes les trahisons volontaires ou involontaires du transporteur (des procédures de communication) et des ruses imaginables

Le passeport biométrique attaqué

Les passeports biométriques français contiennent des puces dites RFID capables de communiquer sans fils. Les données personnelles du voyageur qu'elles contiennent ne deviennent accessibles qu'après que le lecteur de passeport a prouvé son droit de les recevoir en lisant, puis en transmettant à la puce, la clef chiffrée encodée en bas de la première page du passeport. Une fois cette formalité accomplie, le lecteur et le passeport créent une nouvelle clef temporaire pour chiffrer le numéro de passeport, le nom, la date de naissance, la nationalité et la photographie du propriétaire du document.

Toutefois, ce système de sécurité s'est révélé vulnérable. En 2005, il est apparu qu'il était facile de « casser » la clef de sécurité en employant la méthode dite de la force brute. Des failles logiques ont aussi été découvertes dans la conception du système. La première permettait de reconnaître le voyageur en exploitant un message d'erreur envoyé au cours de l'authentification.

En 2010, elle a été corrigée, mais on s'est alors aperçu que la durée exacte du temps de réponse livrait aussi l'identité du voyageur… Plus récemment encore, nous avons montré que la taille et le format des données chiffrées suffisent pour différencier de nombreux passeports, et donc pour les reconnaître dans un contexte particulier. Toutes ces attaques exigent toutefois de savantes mises en œuvre : le contenu des puces RFID de nos passeports est bien mieux protégé que celui de la plupart des puces RFID utilisées au quotidien…

Véronique Cortier,
CNRS-LORIA, Nancy

© Cyril Houshuterstock.com

par les bandits (par les pirates informatiques) ; il nous reste à passer à la pratique (à implémenter le système). Dans cette étape aussi, des failles de sécurité peuvent passer inaperçues. Les spécifications des protocoles n'étant jamais assez exhaustives, il est fréquent que les interprétations qu'en font ceux qui les implémentent confèrent au système certaines particularités, des incompatibilités avec d'autres systèmes, voire des failles de sécurité…

En 2008, on a ainsi découvert une faille dans la version de la suite logicielle *Open*SSH utilisée sur les serveurs *Linux* tournant avec le système d'exploitation *Debian*. Cette suite a été développée par l'équipe du projet *Open*BSD (un projet d'environnement UNIX libre) pour faciliter aux programmeurs l'établissement de connexions sécurisées à partir du protocole SSH (*Secure Shell* en anglais, ou ligne de commande sécurisée). Or il s'est avéré que les paires de clefs RSA de 2048 bits engendrées à l'aide de la version *Debian* de la bibliothèque d'outils de chiffrement *Open*SSL (utilisée par la suite logicielle *Open*SSH) entre 2006 et 2008 avaient en réalité une robustesse équivalente à celle d'une clef de... 16 bits. De fait, 20 minutes suffisaient pour casser les cryptages SSH pour serveurs *Debian*, et prendre ainsi le contrôle de la machine…Compromises, toutes ces clefs ont dû être changées.

Un autre exemple est celui du passeport électronique français introduit en 2006. Ce passeport incorpore une puce de radio-identification (ou RFID, pour *radio frequency identification*) qui, dans la version biométrique en circulation depuis 2008, contient les photos du détenteur et de deux de ses empreintes digitales. Si les spécifications de l'un des protocoles de communication entre la puce RFID du passeport et la borne de lecture étaient initialement correctes, l'implémentation choisie pour la version française du passeport introduisait une faille. Ses concepteurs avaient prévu qu'en cas d'incident d'authentication du passeport auprès d'une borne, un message d'erreur serait émis pour en préciser la cause.

L'intention était bonne, mais cette circulation supplémentaire d'information s'est révélée suffisante pour qu'un attaquant interceptant les communications puisse identifier le passeport français en voie d'authentification, et ainsi le suivre à la trace à travers le monde *(voir l'encadré ci-contre)*.

Ce type d'attaque ne remet pas en cause la spécification du système et ne concerne que certaines implémentations. Il est donc généralement plus aisé de les bloquer. Bien entendu, le mieux est toujours de les éviter en amont de l'implémentation. Il s'agit, là encore, de tester le système de sécurité en le simulant par un modèle théorique le plus réaliste possible. Malheureusement, tout modèle reste… un modèle, c'est-à-dire un jeu permettant de découvrir des attaques possibles dans un cadre restreint, parce que théorique…

Les failles non encore rendues publiques se monnayent

Notons que les failles dont l'existence n'a pas encore été rendue publique se vendent à prix d'or. Il existe en effet un marché de ce que l'on nomme des vulnérabilités « jour zéro » (le jour 1 étant celui de la publication). Ces attaques restées secrètes sont vendues au plus offrant, le plus souvent à des entreprises ou à des États concernés.

Quand la sécurisation des données repose sur une primitive inviolable, qu'elle a été conçue sans aucune faute logique et implé-

BIBLIOGRAPHIE

T. Chothia et V. Smirnov, **A traceability attack against e-passports**, dans *Financial Cryptography*, Springer, vol. 6052, pp. 20-34, 2010.

A. Armando *et al.*, **Formal analysis of SAML 2.0 web browser single sign on : Breaking the SAML-based single sign-on for Google apps.**, 6th ACM Workshop on Formal Methods in Security Engineering (FMSE 2008).

A. K. Lenstra, **Key Lengths, Contribution to the Handbook of Information Security**, Université d'Eindhoven, 2004.

SUR LE WEB

Page Internet du LORIA sur le passeport électronique :
www.loria.fr/~glondu/epassport/attaque-tailles.html

Cours et séminaires du Collège de France sur la sécurité informatique :
www.college-de-france.fr/site/martin-abadi/