Guide for the attention of developers/hosts for merchant websites on the minimum level of security for bank card data processing

Foreword

This guide in no way intends to replace a PCI DSS certification process.

The purpose of this guide is to supply developers/hosts of online payment solutions with information on best practices for implementing the "*minimum security measures*" required to counter the simplest attacks most frequently used by external attackers.

Introduction

Numerous "small/medium-sized" merchants that wish to develop "*e-business*" services use service providers for the development and/or hosting of a payment solution. Because the developers /hosts of e-commerce sites are indirect players in the card-payment chain, their security level forms an integral part of the overall strategy for the protection of bank card data.

A solution recommended by the *Groupement des Cartes Bancaires* is to de-correlate the commercial transaction function from the bank card payment function. This second function requires very high protection due to the sensitivity of the card data. One of the simplest means of doing this is to make use of a specialised e-payment contractor who is already PCI DSS certified. In this way, the merchant and their host, providing that they do not also retain the bank card data, are exempt from another PCI DSS certification.

Nevertheless, if the merchant chooses to develop the e payment function themselves or have it developed by a contractor, they must begin a process to bring their information system into compliance with the requirements of PCI DSS. Depending on the volumes of bank card transactions that they process per year, determining their level' of activity (4 levels) they must comply with the specific requirements attached to each level determined by the international Payment Schemes. This guide is addressed to all levels merchants.

These contractors, web sites developers, etc. do not always have good knowledge of the PCI DSS certification process. Remember that it aims to check compliance with the protection measures, for an information system that stores, transfers and processes bank card data, with the detailed requirements of PCI DSS (about 300 points of control).

Yet, it appears during audits that, in many cases, web sites do not comply with elementary rules of best practice for the development or hosting of secured web systems (Top 10 of website vulnerabilities [1]).

This guide is also intended to supply merchants with elements for discussion and points of control concerning the offers that may be made to them by their contractors.

Contextual elements

In 2011, the annual report from Verizon [2] stated that 92% of breaches came from external attacks. This figure is strongly up (+22%). The reason is not due to an increase in the number of hackers (ultra-specialists), but due to their easy and free access of "*off-the-shelf*" attack software, usable by non-specialists.

In half of the cases, these attacks use basic faults in systems, particularly those of web sites. The finding was that in 92% of cases the attacks were **simple to carry out** and based on **vulnerabilities that had been identified for a very long time**. It is revealing that 96% of breaches could have been avoided by applying relatively simple best-practice security measures.

These are the same observations mentioned by Patrick Pailloux, the director of the French national agency for the security of information systems (ANSSI) in his closing speech[1] to the *Assises de la Sécurité 2011*, where he said "*[…] The reality, that we see every day, is that our information systems are very often permeable, and very often, malicious players have taken full advantage of this.* ", Adding "*we must recover power over our own systems* ".

Best practices reference frameworks

Several reference documents for the community of developers on the web give detailed descriptions of the simplest attacks that use programming vulnerabilities of web sites. The most well-known is the "*Open Web Application Security Project* ", better known under its acronym of OWASP. It regularly updates the classification of the 10 most commonly-used vulnerabilities for compromising websites and accessing information systems that host them and subsequently accessing the data that is stored there.

Numerous well-publicised thefts of several million bank card accounts are based on this type of attack.

More recently, the report [3] from the British "*Centre for the Protection of National Infrastructure (CPNI)*" gives advice on the best ways of protecting against the attacks that are most commonly used by hackers. This last report consists of elementary and common sense advice. Implementing this advice considerably reduces the attack surface visible by attackers.

Of course, one should focus on PCI DSS requirements [4], which are not a guide to best practice in itself, but a reference framework that allows developers to become aware of the points of control that the auditors will check during the PCI DSS certification process.

Lastly, we strongly encourage the various players to consult the guides to best practices for securing information systems on the french government's ANSSI site [5].

[1] http://www.ssi.gouv.fr/fr/anssi/publications/discours-de-patrick-pailloux-lors-de-la-conference-de-cloture-des-assises-de-la.html

The most frequently used vulnerabilities

This chapter does not aim to repeat the advice of the OWASP in which software developers will find the technical elements relevant to the implementation of security measures. Rather, it is aimed at merchants or contractors in charge of developments.

The attack by injection (injection flaw)
This consists of using an entry zone for a search criterion in a web form to "*inject*" executable code. Instead of entering their search criteria (for example, an account number), the attacker enters the characters of a logical formula (e.g.: 'or'1'='1). The software that analyses the entry zone will "*interpret*" the characters by executing the command.

The attack by cross-site scripting (XSS)
This consists of using the entry zones on a web page (e.g.: surname, forename, etc.), via the browser to force the execution of an executable command (script). This attack can usurp a user's session (man in the middle), disfigure a website or steal the identities of legitimate users.

Breach of authentication management
This attack uses a poor implementation of the authentication functions on a web site. The attacker may compromise the passwords or session cookies to steal the access permissions of legitimate users (and therefore their privileges).

Direct non-secured access to an object
This attack can occur when a developer references an object that he/she is handling by using its explicit name (file, database record, etc.). The attacker can then manipulate these references to access the data that is exposed.

Cross-site request forgery (CSRF)
This attack forces the browser belonging to a correctly-authenticated victim to send a false HTTP request to another web application that is known for its vulnerabilities. The target application identifies the request as coming from a legitimate user.

Poor security configuration
The attack consists of using the fact that the software used for developing the web site is not in its latest updated version and that there are known vulnerabilities in the versions used.

Low-security cryptographic storage
On a web site, the means of protecting sensitive data use weak algorithms or keys. The attackers may copy the data and subsequently decrypt it.

Unrestricted URL access
A user who is authorised on certain web pages can dynamically change the name of the URL to access pages that normally require authentication. Certain applications do not sufficiently restrict access to sensitive pages.

Insufficient protection of the transport layer

The attack consists of monitoring and listening to users' network traffic. This relates to web sites for which the authentication is not protected by HTTPS during the authentication phase. The identifiers and passwords are sent in plain text and their interception may lead to identity theft.

Redirection and invalid URL forwarding

The attacker creates links to non-valid redirection URLs (phishing sites or malicious software sites) and invites users to click on them. The victims are inclined to click on these links because they appear to point to a valid site. The attacker uses unsafe forwards to bypass the security checks.

Elementary best practices

Concentrate on the essential checks: many companies make the mistake of aiming at a very high level of security on certain points while completely neglecting others. One is much better protected when elementary standards are applied throughout the entire organisation.

Do not keep useless data: if you don't need this data, do not keep it. On the other hand, it is imperative that useful data is identified, monitored and stored safely.

Secure remote access services: restrict permissions to predefined networks and IP addresses to limit public access. Companies should also restrict access to sensitive information within their internal networks.

Monitor the accounts of employees who have the most extensive permissions: the best approach consists of trusting the user after having verified their probity during the hiring process, granting them limited access permissions according to their role or responsibilities. Managers must give clear security instructions and check that staffs actually do comply with the rules and procedures in place.

Regularly check user accounts: make sure that active accounts are valid, useful, correctly configured and that the corresponding access permissions are adequate (the least permissive possible). Make sure that inactive accounts, and accounts of employees who have left, are deleted.

Manage and analyse the logs: this does not mean studying them line by line, but taking an interest in the main problems. The important point is to reduce the period for the detection of infractions to a few days only. For a best data protection, prefer the monitoring and alert processes that are the most intelligent, effective and responsive.

Educate employees concerning methods of social engineering and the various attack vectors that they represent: teach them to be careful before clicking on links and to mistrust attachments in e-mails where they do not know the sender.

How can you check your level of vulnerability?

Vulnerability and penetration tests: this is certainly the first thing to be done to learn the current situation and what needs to be done to improve it. A vulnerability test can be automatic and performed remotely from tools that are on the market. Penetration tests are carried out by experts who use their targeted expertise of the particular environment that they are analysing.

As required by PCI DSS, vulnerability tests must be carried out every quarter to check that the changes that are regularly made to systems have not increased the level of vulnerability to external attacks.

The code audit: this measure may be necessary to analyse the applications that already exist and that have been shown to be vulnerable during the above-mentioned tests. The technical guides from OWASP allow the performance of code audits and the correction of vulnerabilities.

The education and training of developers: of course, the best way of not having vulnerabilities in an application is to program it according to best practice for good security. There again, guides exist, such as the "*OWASP Developer's Guide* ". This does not mean that it is not necessary to perform regular vulnerability tests.

The implementation of corrective measures

Once the vulnerabilities have been properly identified, the company may use the approach by priorities [6] suggested by the PCI SSC to plan a project to implement the corrective measures.

Summary

The security of bank card data concerns all players without exception, including hosts. Remember that the PCI DSS requirements apply to every entity that stores, processes or transmits sensitive data.

Once again, 92% of observed attacks are technically simple. Consequently, the systematic implementation of basic protection and checks that are well-known to the IT world and heads of information-systems security is an imperative prerequisite for the proper protection of sensitive bank card data.

References

[1] OWASP Foundation, "*OWASP Top 10 – 2010 The ten most critical security risks for web applications*, (Open Web Application Security Project)"

[2] Verizon, "2011 Data Breach Investigation Report"

[3] Centre for the Protection of National Infrastructure; "*Development and implementation of secure web applications*", August 2011 (British national organisation in charge of the protection of the country's critical infrastructure – In France, these duties are performed by the ANSSI)

[4] Payment Card Industry – Data Security Standard, *Requirements and Security Assessment Procedures*, Version 2.0, October 2010

[5] The link towards the ANSSI site: http://www.ssi.gouv.fr/fr/bonnes-pratiques/

[6] Payment Card Industry – PCI DSS Prioritized Approach for PCI DSS 2.0, May 2011