

THE PROBLEM

The client wanted to test their product's compliance with the "Payment Card Industry Data Security Standard (PCI DSS)". This is a widely accepted set of policies and procedures intended to optimize the security of credit, debit and cash card transactions and protect cardholders against misuse of their personal information.

In this Case study we discuss our systematic examination (often known as peer review) of the source code that was intended to find and fix issues that were non compliant with PCI DSS standards in the development phase, hence improving overall security and quality of software.

The client's requirement was to check the quality of ASP.Net based code against the PCI standard, which was also our basic mechanism for validating the design and implementation of patches. It also helped us maintain a level of consistency in design and

THE APPROACH

Our primary approach was to gather sufficient domain knowledge in order to mark areas that required enhancements. The QA team at Kualitatem developed a checklist based on the combination of PCI standards and the critical requirements outlined by the client. This checklist was developed to verify the code written in ASP.Net language.

Code review is a rather agonizing experience for all involved, particularly the design and development team. The QA team at Kualitatem is adept at presenting code reviews as part of enhancing and upgrading the application without giving it a flavor of criticism. We keenly focus on the following points while carrying out code reviews:

- Ask questions rather than make statements.
- Avoid the "Why" questions.
- Remember to praise.
- Ensuring good coding standards and best practices to reference.
- Make sure the discussion stays focused on the code and not the coder.
- Remember that there is often more than one way to approach a solution.

After developing the checklist our testing team reviewed the code and analyzed the factors important for the proper code review. Review was done after proper analysis, and following steps were performed to review the code successfully.

For the proper code review our QA team developed the checklist of the proposed statements and then after reviewing the code made a report in the form of checklist which is as follow as sample.

Sr#	Steps
1	Remove all unnecessary functionality
2	Encrypt all non-console administrative access
3	Encrypt transmission of cardholder data across open, public networks
4	Develop and maintain secure systems and applications
5	Input not validated
6	Broken Access Control
7	Broken Authentication and Session Management
8	Cross Site Scripting(XSS) Flaws
9	Buffer Overflow
10	Injection Flaws
11	Improper Error Handling
12	Insecure Storage
13	Denial of Service
14	Insecure Configuration Management

Table 1

PCI Requirement:	Remove all un-necessary functionality
Review:	YES
Compliance:	No vulnerability exists under this category
Recommendations:	NIL
PCI Requirement:	Encrypt all non-console administrative access
Review:	YES
Compliance	No vulnerability exists under this category
Recommendations	SSL web interface should be used for cryptography

Table 2

SUMMARY

Credit card based business applications are attractive and vulnerable targets for hackers. Testing such an application demands more vigilance and scrutiny from the test team.

In case of this product where PCI compliance was to be tested, the checklists were used to carry out meticulous analysis and review of the product code to check for security loopholes. As the product was credit card based, security was a critical factor while gauging quality. The team also put the code through a 'penetration test, to discover ways in which hackers and unauthorized users can penetrate the system. '. All issues reported through the testing process were reported to the client, hence enabling them to deliver a secure and stable product to the end user.