



## Code Review of ASP.Net Using PCI Standards

### Response

Primary task in any testing activity is to gather the domain knowledge.

KRAYSIS QA team developed a check list based on the PCI standards and according to the requirements of client. This check list was developed to verify the code written in ASP.Net language.

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

### Background

The focus of the client is 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. The aim is systematic examination (often known as peer review) of the source code intended to find and fix issues of not following PCI standards in the development phase, improving the overall security and quality of software.

### Challenge

- Primary challenge is to make quality Code Review of ASP.Net product as per PCI standards that is our basic mechanism for validating the design and implementation of patches. It also helps us maintain a level of consistency in design and implementation practices across various modules.

Gathering domain knowledge to suggest improvements.



| Sr. # | Description  |
|-------|--|
| 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 Overflows   |
| 10    | Injection Flaws  |
| 11    | Improper Error Handling  |
| 12    | Insecure Storage   |
| 13    | Denial of Service  |
| 14    | Insecure Configuration Management                                    |

### Process

Code review has always been a painful experience for everyone involved. The developer often takes it as an unhealthy criticism against their will. The development leads are often confused as to what is important to point out and what isn't. And other developers that may be involved often use this as a chance to show how much better they can be by pointing out possible issues in someone else's code.

But Kraysis QA Team used different approach.

- 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.

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,

|                  |   |
|------------------|---|
| PCI Requirement: | Remove all unnecessary functionality        |
| Review:          | YES   |
| Compliance:      | No vulnerability exists under this category |
| Recommendation:  |   |

|                  |  |
|------------------|--|
| PCI Requirement: | Encrypt all non-console administrative access      |
| Review:          | YES  |
| Compliance:      | No vulnerability exists under this category        |
| Recommendation:  | SSL web interface should be used for cryptography. |

|                  |  |
|------------------|--|
| PCI Requirement: | Encrypt transmission of cardholder data across open, public networks   |
| Review:          | YES  |
| Compliance:      | No vulnerability exists under this category  |
| Recommendation:  | <ol style="list-style-type: none"> <li>1. Sensitive information must be encrypted during transmission over open, public network that are easy and common for hackers to intercept, modify and divert data while in transit.</li> <li>2. Use strong cryptography and encryption techniques (of at least 128 -bit) such as Secure Socket Layer (SSL)</li> <li>3. Use S/MIME encryption and digital signature technology</li> </ol> |

|                  |   |
|------------------|---|
| PCI Requirement: | Buffer Overflows                            |
| Review:          | YES   |
| Compliance:      | No vulnerability exists under this category |
| Recommendation:  |   |

|                  |  |
|------------------|--|
| PCI Requirement: | Injection Flaws  |
| Review:          | YES  |
| Compliance:      | No vulnerability exists under this category                                  |
| Recommendation:  | All input field should be validated for valid format of required parameters. |

Kraysis QA team used the above checklist to improve the long-term outlook for code quality of the product by proper analysis of the code.

## Conclusion

KRAYSIS provided seamlessly integrated software quality assurance services using innovative framework and standards.

Kraysis team scrutinized the code according to the prevailing PCI standards. Focus of this code review was to help ensure PCI compliance for the client and also to look for security loopholes in the code. Being a credit card based product, security became an important factor to judge the quality of the product. Along with security, other measures were also required by the client to go through in order to affirm better quality of code. Kraysis did incorporate all these requirements of the client in the process and pointed issues which were then resolved by the client.

### United Kingdom

Kraysis Ltd Corporate Headquarters,  
139 - Groveway, Dagenham, Essex,  
RM8 3XL, United Kingdom.  
Voice: +44 203 287 1186

### Pakistan

Suite # 415, Al-Qadir Heights,  
1 Babar Block, New Garden Town, 54700,  
Pakistan.  
Voice: +92 42 32404824-25