

Countermeasures Theoretical Cost-Effectiveness

This report produces a list of countermeasures sorted by their theoretical cost-effectiveness, based on the assumption that all countermeasures will be implemented. For each countermeasure, the report displays calculative parameters such as cost-effectiveness, implementation cost and the overall mitigation level of the specific countermeasure. It also displays a list of the vulnerabilities mitigated by each countermeasure.

C022 Do not provide any server or system error code information

Description:

Do not pass the MS SQL exceptions to a WebUser without filtering. Replace msg with "Please contact the system admin"

Cost Effectiveness: 80.3 % per 1,000 \$

Implementation Cost: 500 \$

Overall Mitigation: 40.2 % of total risk

Mitigated Vulnerabilities:

V016 Accidental leaking of sensitive information through error messages

C006 Implement validation of input fields in online order form

Description:

For example: validate input query string in order query page. The cost expresses the one time effort for developing this software feature.

Cost Effectiveness: 61.8 % per 1,000 \$

Implementation Cost: 1,000 \$

Overall Mitigation: 61.8 % of total risk

Mitigated Vulnerabilities:

V006 Online order page enables SQL injection via un-validated query string

C001 Don't use default administrator password, use generated passwords

Description:

Use of generated passwords which are changed automatically and must be entered at given time intervals by a system administrator. These passwords will be held in memory and only be valid for the time intervals. Login messages from client to backend serverd sent should be tagged and checksummed with time sensitive values so as to prevent replay style attacks.

Cost Effectiveness: 11.4 % per 1,000 \$

Implementation Cost: 4,500 \$

Overall Mitigation: 51.3 % of total risk

Mitigated Vulnerabilities:

V001 Hard coded password

C021 Implement key exchange with entity authentication

Description:

Key exchange without entity authentication may lead to a set of attacks known as "man-in-the-middle" attacks. These attacks take place through the impersonation of a trusted server by a malicious server. If the user skips or ignores the failure of authentication, the server may request authentication information from the user and then use this information with the true server to either sniff the legitimate traffic between the user and host or simply to log in manually with the user's credentials.

Cost Effectiveness: 8.0 % per 1,000 \$

Implementation Cost: 5,000 \$

Overall Mitigation: 40.2 % of total risk

Mitigated Vulnerabilities:

V001 Hard coded password

C005 Database login accounts should be given the minimal rights that are necessary for their functionality

Description:

Web application account used for retrieving daily rates is assigned with read only permissions. Admin account is given update privileges only on FC rates data. DB administrator is the only account with full rights on the database that can access and modify data. The cost reflects administration effort.

Cost Effectiveness: 3.8 % per 1,000 \$

Implementation Cost: 3,500 \$

Overall Mitigation: 13.3 % of total risk

Mitigated Vulnerabilities:

V009 Command line SQL can be used to view/modify sales system tables

C019 Security officer will have mandate to assure the personal integrity of contract DBA

Description:

Cost Effectiveness: 2.7 % per 1,000 \$

Implementation Cost: 5,000 \$

Overall Mitigation: 13.3 % of total risk

Mitigated Vulnerabilities:

V013 Personal weaknesses of DBA may be exploited by hostile parties

C007 Enforce data access via stored procedures with formal parameters content validation

Description:

Data in database should be manipulated only via stored procedures. The parameters of the stored procedures should be validate for their content before executing the stored procedure. The cost here is the one time effort for developing this software feature.

Cost Effectiveness: 2.2 % per 1,000 \$

Implementation Cost: 10,000 \$

Overall Mitigation: 21.6 % of total risk

Mitigated Vulnerabilities:

V006 Online order page enables SQL injection via un-validated query string