

**Application Threat Modeling**

**eCoaching Log System**

May 29, 2014

**Change History Log**

| **Date** | **Change Description** | **Author** |
| --- | --- | --- |
| 5/29/14 | Initial version | Timothy Queen |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| **Scope** | This *Threat Modeling Report Template* is to be used by GDIT. software projects to document any security risk and necessary controls for mitigation. This document is used by the Project Security Analyst (PSA) as an internal team deliverable to assist with the Security Risk & Requirements. The Threat Modeling approach for Vangent is based on the Microsoft® Threat Model. However, the Vangent approach will be less detailed given the nature of the SDLC and security integration points.  The Software Project Manager (SPM), Project Team, and the Project Security Analyst will collaborate on the Threat Model. The Security Services consultant will use results to perform the further risk analysis and document findings to the SPM in the Security Risk and Requirements. |

|  |  |
| --- | --- |
| **Purpose** | You cannot build secure applications without identifying threats to the application. Once threats are properly identified and ranked the development team can determine a course of action (secure programming) to reduce overall risk to the application. The purpose of the Threat Model is to provide information about application threats to the overall Risk Assessment process and resulting Security Risk & Requirements document. This information will ultimately find its way into the project’s requirements (Functional Specification or Functional Design).  A Threat Model will allow collaboration among team members and ultimately provide simple view of the threats to the application and ways to mitigate each threat. The goal is to reduce the overall risk to the application. This document will provide a threat model by evaluating each of the areas outlined below:   * Identify Assets * Architectural Overview & Decomposition * Identify & Rate Threats * Choose how to respond threats & techniques to mitigate |

**Identify Assets**

List all assets within the applications production environment. This would be any asset that would be significant enough to require protection of its confidentiality, integrity, and availability.

|  |  |  |
| --- | --- | --- |
| **Assets** | **Description** | Comment |
| *Hardware Components* | *Physical hardware /virtual environment where the application resides.* | *More important than the actual hardware is the software components stored on disk. Falls under standard security for all hardware.* |
| *VADENMWBP11* | *Web Server* |  |
| *VDENSSDBP07\SCORP01,1436* | *Database Server* |  |
| *Software Components* | *Intellectual Property. Assemblies, logs, configuration. server software, database, etc.* | *Software components (specifically interpreted language components; Java, .Net) can be reverse engineered to reveal intellectual Property, trade secrets, etc. Can also reveal information that can be used to further compromise IT resources.* |
| *Windows 2003, 2008 Server* | *Operating System* |  |
| *Microsoft IIS* | *Web Server* |  |
| *Microsoft SQL Server 2008* | *Database Software* |  |
| *ASP.NET, c#.NET, ADO.NET* | *Software used to write the solution* |  |
| [*http://cwe.vangent.local/sites/teams/CMS/analytics/Performance\_Scorecard/default.aspx*](http://cwe.vangent.local/sites/teams/CMS/analytics/Performance_Scorecard/default.aspx) | *Performance Scorecard SharePoint launch site* | *Links for the Performance Scorecard and Alternate Channels Scorecard can be found at this link.* |
| *Database log files* | *Log files* |  |
| *.dbf* | *Database files* |  |
| *Network* | *Network and its traffic data.* | *The application runs on a networked server with connectivity to internal network and its resources.* |
| VLAN | 10.24.x.x | Network assets are beyond the scope of the modeling document |
| Core | 10.30.x.x | Network assets are beyond the scope of the modeling document |
| LDAP Server | Domain Controller | Network assets are beyond the scope of the modeling document |

**Architectural Overview**

A complete description of the architecture can be found in the Architecture Notebook.

******

|  |  |
| --- | --- |
| **Entry Point** | **Usage Scenario** |
| CSR | * Authenticates to Active Directory * Views their scorecard information |
| Supervisor | * Authenticates to Active Directory * Views their team information |
| Manager | * Authenticates to Active Directory * Views their team information |

**Technologies**

* Windows Server 2003, 2008
* MS SQL Server 2005/8
* Microsoft Windows 7
* Internet Explorer 9+
* IIS
* SSL
* TCP/IP tools
* ASP.Net
* C#.net

**Application Security Mechanisms**

* Input and data validation
  + Drop down lists
  + Check boxes
  + Radio buttons
  + Obfuscation
* Authentication
  + Users are authenticated using their domain credentials
  + Service accounts are used for database access
* Authorization
  + Users are assigned a role within the application
  + Roles are granted access to parts of the application that are needed to perform that role
  + Least privilege configuration on the servers, database and applications
* Configuration Management
  + Configuration is managed by the Performance Scorecard team
  + Configuration items are stored in Serena Version Manager
  + Configuration management processes are used for implementing approved changes
* Session Management
  + Session timeouts
* Cryptography
  + SSL
  + IPSec
* Parameter Manipulation
  + Passing of parameters between pages is limited
* Exception Management
  + Periodic security vulnerability scans
* Auditing and Logging
  + System logs capture logs generated by the OS
  + Database logs capture database transactions
  + The application writes to system logs capturing significant events
  + Audit logs are created for activity within the application

# Decompose the application

*A complete list of the applications used in the system can be found in the EDD for this system.*

**Threats**

All threats are associated with a security property.

|  |  |
| --- | --- |
| **Threat** | **Security Property** |
| Spoofing | Authentication |
| Tampering | Integrity |
| Repudiation | Nonrepudiation |
| Information disclosure | Confidentiality |
| Denial of service | Availability |
| Elevation of privilege | Authorization |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **DFD Element** | **Spoofing** | **Tampering** | **Repudiation** | **Information Disclosure** | **Denial of Service** | **Elevation of Privilege** |
| Data Flows |  | X |  | X | X |  |
| Data Stores |  | X |  | X | X |  |
| Processes | X | X | X | X | X | X |
| Interactors | X |  | X |  |  |  |

The Threat identification, documentation, and ratings are combined in the table below. Stride Model used for threats and DREAD for ratings.

|  |  |
| --- | --- |
| Use the **STRIDE** model to indicate the types of threats. | For each threat provide a rating using **DREAD**.  DREAD RATING = (D + R + E + A + D) / 5 |
| S Spoofing  T Tampering  R Repudiation  I Information Disclosure  D Denial of Service  E Elevation of Privilege | D Damage potential (Worst is 10. Example: Elevation of privilege is 10)  R Reproducibility (Turning threat into exploit? Works every time then 10)  E Exploitability (How hard? If a novice can perform attack then 10)  A Affected users (How many? 91-100 percent is 10. 0-10 percent is 1)  D Discoverability (Assume anyone will discover threat and exploit it.  So score is 10 unless really obscure exploit) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Threat**  (Note: Numbered items refer to items in the graphic above) | **Threat Types** | **DREAD SCORE** | **DREAD RATING** | Comments / Recommendations | Response |
| *eCL feed data flow:* eCL Feed may be spoofed by an attacker and this may lead to incorrect data delivered to Load. Improper data protection of eCL Feed can allow an attacker to read information not intended for disclosure. | *SI* | *D-5*  *R-3*  *E-3*  *A-5*  *D-10* | *5.2* | * *Use standard authentication* * *Review authorization settings* | * *All users authenticate against active directory* * *Only authorized users may access the data and the server* |
| 1. *Web server to browser:* If Web Server is given access to memory, such as shared memory or pointers, or is given the ability to control what Browser Client executes (for example, passing back a function pointer.), then Web Server can tamper with Browser Client. Packets or messages without sequence numbers or timestamps can be captured and replayed in a wide variety of ways. | *TD* | *D-3*  *R-3*  *E-3*  *A-1*  *D-10* | *4* | * *Do not use pointers* * *Use standard protocols* | * *Program does not pass pointers* * *HTTPS is used for communications* |
| 1. Data load to database: SQL Database may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of SQL Database. SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives. Excessive resource utilization could result in a denial of service. | STID | D-5  R-5  E-3  A-3  D-10 | 5.2 | * Use standard authentication techniques * Review SQL for possible SQL insertion * Monitor resource usage | * Users are authenticated against Active Directory * Text input in scanned for SQL statements * The NOC monitors resource usage. |
| 1. Outlier data flow: Outliers may be spoofed by an attacker and this may lead to incorrect data delivered to Load. Improper data protection of Outliers can allow an attacker to read information not intended for disclosure. | SI | D-5  R-5  E-3  A-3  D-10 | 5.2 | * Use standard authorization * Review authorization settings | * Users are authenticated against Active Directory * Only authorized users can access the data |
| 1. Database to web server: SQL Database may be spoofed by an attacker and this may lead to incorrect data delivered to Web Server. Improper data protection of SQL Database can allow an attacker to read information not intended for disclosure. | SI | D-5  R-5  E-3  A-3  D-10 | 5.2 | * Use standard authorization * Review authorization settings | * Users are authenticated against Active Directory * Only authorized users can access the data |
| 1. Web server to database: SQL Database may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of SQL Database. SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives. Excessive resource utilization could result in a denial of service. | STID | D-5  R-5  E-3  A-3  D-10 | 5.2 | * Use standard authentication techniques * Review SQL for possible SQL insertion * Monitor resource usage | * Users are authenticated against Active Directory * Text input in scanned for SQL statements * The NOC monitors resource usage. |
| 1. User data to Load: User Data may be spoofed by an attacker and this may lead to incorrect data delivered to Load. Improper data protection of User Data can allow an attacker to read information not intended for disclosure. | SI | D-5  R-5  E-3  A-3  D-10 | 5.2 | * Use standard authorization * Review authorization settings | * Users are authenticated against Active Directory * Only authorized users can access the data |

# References

How To Create a Threat Model: <http://msdn.microsoft.com/en-us/library/ff647894.aspx>

Threat Modeling, Frank Swiderski and Window Snyder

The Security Development Lifecycle, Michael Howard and Steve Lipner