D7.2: Global threat model analysis document

Lead contractor: S2 Grupo
Author(s): David Monteagudo, Roberto Amado, Daniel De Los Reyes, Sergio Zamarripa
Submission date: June 2011
Dissemination level: Public

Project co-funded by the European Commission under the Seventh Framework Programme
© S2, TUM, LERO, INESC-ID, TXT, PRODEVELOP
Abstract: This document describes all the identified threats, establishing a taxonomy of threats that will be compared to real data obtained in similar environments.
Contents

1 Introduction 5
  1.1 Definitions ........................................ 5

2 The Security Development Lifecycle 7
  2.1 OWASP Approach .................................. 7
  2.2 Microsoft’s Approach to Secure Software Development .... 8
    2.2.1 Requirements ................................... 9
    2.2.2 Design ........................................ 9
    2.2.3 Implementation ................................ 9
    2.2.4 Verification ................................... 10
    2.2.5 Release ....................................... 10

3 What is Threat Modelling 11

4 Available Tools 12
  4.1 Microsoft Threat Modeling Tool .......................... 12

5 Threat Modelling in FastFix 14
  5.1 Motivation ......................................... 14
  5.2 Actions taken ...................................... 14

Bibliography 17

6 Attachments 18
List of Figures

2.1 Microsoft SDL Lifecycle ................................................. 8
4.1 Simplified Microsoft SDL Process ................................. 12
5.1 Main conceptual architecture elements ............................ 15
5.2 FastFix Core ............................................................. 15
1 Introduction

Application security has become a major concern in recent years. Access to sensitive data or application disabling are simple examples of common goals for hackers worldwide, and are the reason that explain the imperative need to secure any application in today’s world.

It’s not been so long that application security was considered, whenever this happened, only as the last phase at the end of development cycle, while developers focused on functionality and features most of their time. This approach has been identified to be really dangerous and extremely expensive, as many vulnerabilities are never detected until in a production environment after the application has been attacked and damaged.

This is one of the motivations to incorporate security throughout the software lifecycle development, particularly in the early stages of this process. In order to follow such paradigm, software engineering methodologies and practices need to be developed that support the simultaneous analysis of both security and software requirements, in order to meet this need arises threat modelling.

Threat modeling is essentially a risk analysis technique oriented to applications, that allows developers to analyze different environments threats to their systems, thus optimizing the allocation of resources to those parts of the system that need it most, and to focus on the weak points of the application.

Among the advantages, it should be noted that this approach allows great flexibility and the necessary conditions to acquire a practical overview of the system, not being conditioned by a defined set of threats and disaster scenarios, as in the classical risk analysis. However, a good threat model is not an unequivocal guarantee of success from the point of view of security and does not necessarily imply the development of secure software, because it suggests patterns, possible error situations, and procedures to improve development practices, but does not involve the obligation to use them in applications, and thus it should be taken as a starting point that must be considered and evaluated regularly to ensure that it is being applied correctly.

Finally, it should be noted that although the application of the model follows an iterative approach and can be applied at any stage of software development, in order to achieve greater effectiveness in the threat modeling process it should be conducted in very early stages of software lifecycle development. This reduces the likelihood of technical, operational and management errors being transferred to the final stage of product, thus becoming vulnerabilities that could be exploited in a production environment.

1.1 Definitions

- **Asset**: any item, whether logical or physical, material or inmaterial, owned by an individual or corporation that is needed to function properly and achieve the objectives.

- **Event**: occurrence happening at a determinable time and place, with or without the
participation of human agents. It may be a part of a chain of occurrences as an effect of a preceding occurrence and as the cause of a succeeding occurrence.

- **Attack**: set of actions aimed at compromising any safety aspects (i.e., integrity, confidentiality or availability).

- **Threat**: is an undesired event that can potentially trigger damage to assets.

- **Risk**: Probability of occurrence of a threat on a particular asset.

- **Potential safeguard**: procedure or device, hardware or software, which reduces the risk.

- **Intrinsic Risk**: risk that is evaluated before considering safeguards in place.

- **Residual Risk**: remaining risk after implementation of safeguards in place.
2 The Security Development Lifecycle

The software development canonical model focuses on the design, analysis, implementation, testing and release of any application. Thus, the project’s goal is to meet all the identified requirements, leaving the review of security conditions at the end of the project.

This situation can be worsened by the absence of explicit requirements related to security in this classical model. This may lead to the emergence of security problems, in later stages of the development process causing significant increases in the project costs, and it could even cause the need of a complete redesign and reconstruction of a component or the entire system.

Based on current practices, it has been found that companies leverage three distinct strategies to address the security threats and vulnerabilities that are latent in their currently deployed portfolios of application software:

- **Find and fix** – i.e., the use of application vulnerability scanning and penetration testing solutions to identify the security vulnerabilities in the applications currently in production, to be addressed subsequently by the application developers.

- **Defend and defer** – i.e., enhancing the security of applications currently in production through the use of web application firewalls or application-level proxies, to reduce or defer the need for security vulnerabilities to be addressed by the developers.

- **Secure at the source** – i.e., the integration of secure application development tools and practices into the software development lifecycle, to increase the elimination of security vulnerabilities before applications are deployed.

2.1 OWASP Approach

The Open Web Application Security Project (OWASP) has its own methodology for threat modeling. In this case, the threat modeling is based on five steps:

- Identification of security objectives or critical assets,
- Analysis of the application
- Decomposition of the application structure
- Identify Threats
- Identifying vulnerabilities
2.2 Microsoft’s Approach to Secure Software Development

As one of the world’s largest software developers, Microsoft has invested heavily in improving the security and privacy of its software and services, with the objective of reducing application security risk for its customers. Dating back to 2004, the Microsoft Security Development Lifecycle (SDL) model has been a company-wide initiative and mandatory policy governing the company’s software development process. By embedding security and privacy throughout its software development lifecycle, Microsoft has also reduced its total cost of development – and generously provided a comprehensive and practical framework that other organizations can leverage for their own application security initiatives.

The SDL is a software development security assurance process that consists of a collection of security practices, grouped by the phases of the traditional software development life cycle. Combining a holistic and practical approach, the SDL introduces security and privacy throughout all phases of the development process, with the goal of optimizing your software security and helping to protect end users. Practical experience at Microsoft has shown that security activities executed in chronological order and as part of a repeatable process result in greater security gains and cost benefits than those resulting from ad hoc implementation. This process is not specific to Microsoft or the Windows platform. It can be applied to different operating systems, platforms, development methodologies, and to projects of any size.

The Microsoft SDL is based on three core concepts—education, continuous process improvement, and accountability. The ongoing education and training of technical job roles within a software development group is critical. The appropriate investment in knowledge transfer helps organizations to react appropriately to changes in technology and the threat landscape. Because security risk is not static, the SDL places heavy emphasis on understanding the cause and effect of security vulnerabilities and requires regular evaluation of SDL processes and introduction of changes in response to new technology advancements or new threats. Data is collected to assess training effectiveness, in-process metrics are used to confirm process compliance and post-release metrics help guide future changes. Finally, the SDL requires the archival of all data necessary to service an application in a crisis. When paired with detailed security response and communication plans, an organization can provide concise and cogent guidance to all affected parties.

This lifecycle, can be split in different phases as shown in next picture.

Figure 2.1: Microsoft SDL Lifecycle
2.2.1 Requirements

The Requirements Phase of the SDL is about specifying the security and privacy requirements of the software to optimize integration of security and privacy during a project. It consists of three practices: Establishing Security and Privacy Requirements, Defining Quality Gates/Bug Bars, and Performing a Security and Privacy Risk Assessment.

Security and privacy requirements analysis includes: Assigning security experts, defining minimum security and privacy criteria for the application as designed to run in its planned operational environment and specifying and deploying a security vulnerability/work item tracking system that allows for creation, triage, assignment, tracking, remediation, and reporting of software vulnerabilities.

A bug bar is a quality gate that applies to the entire software development project and defines the severity thresholds of security vulnerabilities—for example, no known vulnerabilities in the application with a “critical” or “important” rating at time of release. The bug bar, once set, should never be relaxed.

Finally, Security risk assessments (SRAs) and privacy risk assessments (PRAs) identify functional aspects of the software that require closer review.

2.2.2 Design

The Design Phase of the SDL is about defining and documenting the security architecture of the software. It consists of three practices: Establishing Design Requirements, Attack Surface Analysis, and Threat Modeling.

The design requirements activity contains a number of required actions including the creation of security and privacy design specifications, specification review, and specification of minimal cryptographic design requirements.

A thorough analysis will provide better awareness of overall product attack surface. With this information, design considerations should be put in place to reduce attack surface, including but not limited to disabling or restricting access to system services, applying the principle of least privilege, and employing layered defenses where possible.

Threat modeling is a process by which it can be understood security threats to a system, determine risks from those threats, and establish appropriate mitigations. Then for a proper risk assessment and management, the identification and understanding of threats are crucial as they are the starting point.

2.2.3 Implementation

The Implementation Phase of the SDL relates to enforcing security practices to ensure secure software development, and consists of three practices: Using Approved Tools, Deprecating Unsafe Functions/APIs, and Performing Static Analysis.

Define and publish a list of approved tools and associated security checks, such as compiler/linker options and warnings. The list should be regularly updated with the latest versions of the tools.

Determine the list of banned functions, use header files, newer compilers, or code scanning tools to check code for the existence of banned functions, and then replace those banned functions with safer alternatives.
Static analysis consists of analyzing the source code prior to compile and provides a scalable method of security code review and helps ensure that secure coding policies are being followed.

2.2.4 Verification

The Verification Phase of the SDL is about testing the completed software against security and privacy requirements specified at project inception. This phase consists of three practices: Performing Dynamic Analysis, Fuzz Testing, and Attack Surface Review.

Dynamic analysis is run-time verification of your software, leveraging tools which monitor application behavior for memory corruption, user privilege issues, and other critical security problems.

Fuzz Testing is a specialized form of dynamic analysis that induces program failure by deliberately introducing malformed or random data to an application.

Attack surface review is a security practice that ensures any design or implementation changes to the system have been taken into account, and that any new attack vectors created as a result of the changes have been reviewed and mitigated including threat models.

2.2.5 Release


The Incident Response Plan identifies the appropriate points of contact in case of a security emergency. It also includes security servicing plans for code inherited from other groups within the organization and for licensed third-party code. This plan is needed because even programs with no known vulnerabilities at the time of release can be subject to new threats that emerge over time.

The Final Security Review (FSR) is a deliberate examination of all security activities performed on software prior to release. The FSR usually includes an examination of threat models, tools outputs, and performance against the quality gates and bug bars defined during the Requirements Phase. The FSR results in one of three different outcomes: Passed FSR, Passed FSR with exceptions, FSR with escalation.

Certify that the project team has satisfied the security and privacy requirements prior to software release and archive all pertinent information and data, including specifications, source code, binaries, private symbols, threat models, documentation, emergency response plans, and license and servicing terms for any third-party software.
3 What is Threat Modelling

Today’s security management efforts are based on risk management principles. In other words, security resources are applied to vulnerabilities that pose the greatest risk to the business. There are several processes for identifying and prioritizing risk. One of the most effective is threat modeling.

It’s common for security teams to receive reports of vulnerabilities with requests for immediate action to eliminate them. One big source of these requests is an organization’s internal audit team. Another common source of fix-it-now—because-the-press/vendor-says-it’s-critical messages is management.

One of the basic tenets of risk management is that not every vulnerability presents a threat to a network. Only a vulnerability that can be exploited is a threat to business operations and information assets.

Threat modeling is an analysis process that helps to better understand the attack surface of developed components so that it can be understood what is needed to do to ensure that application codification is more secure. It helps to identify those vulnerabilities that are actually critical in the unique environment that is the application network.

The threat modeling process should:

- Identify potential threats and the conditions that must exist for an attack to be successful
- Provide information about how existing safeguards affect required attack conditions
- Provide information about which attack condition and vulnerability remediation activities add the most value
- Help to understand which conditions or vulnerabilities, when eliminated or mitigated, affect multiple threats; this optimizes security investment
4 Available Tools

Because the threat modeling approach is much less common than the approach of classical risk analysis, there are few available tools in the market to assist in this task. However, there is a GUI application offered by Microsoft which has been used for modelling the risks in FastFix.

4.1 Microsoft Threat Modeling Tool

This tool, freely distributed and developed by Microsoft, is one of the most used in this area, with a wealth of information and resources for your use. This tool has its own blog, and a book written by the developers of the application and it's mainly used in the design phase of the SDL.

The SDL Threat Modeling Tool enables any developer or software architect to:

- Communicate about the security design of their systems
- Analyze those designs for potential security issues using a proven methodology
- Suggest and manage mitigations for security issues

It has a guidance system that allows to identify the threats from the application modeling: Once identified, methods to mitigate them have to be indicated and after that, it has to be validated that the safeguards suggested protects against the identified threat.

This process can be explained in a graphical view in the following diagram:

![Figure 4.1: Simplified Microsoft SDL Process](http://blogs.msdn.com/b/threatmodeling)
The SDL Threat Modeling Tool\(^2\) differs from other tools and approaches in two key areas:

It is designed for developers and centered on software: many threat modeling approaches center on assets or attackers. In contrast, the SDL approach to threat modeling is centered on the software. This new tool builds on activities that all software developers and architects are familiar with—such as drawing pictures for their software architecture.

It is focused on design analysis: the term "threat modeling" can refer to either a requirements or a design analysis technique. Sometimes, it refers to a complex blend of the two. The Microsoft SDL approach to threat modeling is a focused design analysis technique.

5 Threat Modelling in FastFix

5.1 Motivation

FastFix is focused on enabling time-and-cost-efficient maintenance and support services, by monitoring software applications, replicating semantic execution failures, and automatically generating patches.

Not only software maintenance and support services are key factors to customers’ perception of software quality, but also security has to be addressed, so software vendors need a system to remotely provide a secure high quality support service to their customers, improve user experience and facilitate corrective, adaptive and preventive maintenance – of both new and existing software products.

Trust is also a key value while monitoring user interaction with a system and collecting context information. As this is one of the requirements of FastFix Platform, it is mandatory that it has a major importance so that users can be confident that it is free of vulnerabilities.

Discovering vulnerabilities is important, but just as important is being able to estimate the associated risk to the business. Early in the life cycle of FastFix platform, security concerns in the architecture or design must be identified using threat modeling.

5.2 Actions taken

For this purpose, Microsoft’s threat modeling process is being used in FastFIX’s development process, following its defined steps: surveying the Application, decompose it, identifying threats, suggesting methods to mitigate them and validating the removal of these threats.

In order to do that, the main conceptual elements of the architecture have been identified as well as the threats to which each of them is subjected, as it can be shown in the following diagram:
Identification of any threat in this platform has been addressed according to a STRIDE model, that allows to identify Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privilege threats.

After this identification at a higher design level, it has been proceeded to conduct a more granular analysis of each of the major building blocks of the platform, identifying its threats as it has been done before.

For instance, the next diagram displays the system from the perspective of the FastFix Core:

The full details of the performed threat model are attached to this document as a convinient method to read it in case Microsoft Threat Modelling Tool is not available. However, the more appropriate way to read the analysis is using the previously mentioned tools since it is an interactive process in which information is linked to each of the entities described in the diagrams. Microsoft’s tool can be freely downloaded from [4] and
FastFIX’s Threat Model file is available for download from the project’s repository.
Bibliography


6 Attachments
## Threat Model Information

<table>
<thead>
<tr>
<th>Component</th>
<th>FastFix Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
<td>FastFix</td>
</tr>
<tr>
<td>SourceUrl</td>
<td><a href="http://www.fastfixproject.eu/">http://www.fastfixproject.eu/</a></td>
</tr>
<tr>
<td>Owner</td>
<td>European Commission</td>
</tr>
<tr>
<td>Participants</td>
<td>S2 Grupo (ES)</td>
</tr>
<tr>
<td></td>
<td>Technische Universitaet Muenchen (DE)</td>
</tr>
<tr>
<td></td>
<td>The Irish Software Engineering Research Center (IE)</td>
</tr>
<tr>
<td></td>
<td>Instituto de Engenharia de Sistemas e Computadores, Investigacao e desenvolvimento em Lisboa (PT)</td>
</tr>
<tr>
<td></td>
<td>TXT E-Solutions SPA (IT)</td>
</tr>
<tr>
<td></td>
<td>Prodevelop (ES)</td>
</tr>
<tr>
<td>Url</td>
<td><a href="http://www.fastfixproject.eu/">http://www.fastfixproject.eu/</a></td>
</tr>
<tr>
<td>Summary</td>
<td>This threat model is focused on identifying all threats and its impact in all elements that act together in fast fix project. It also faces the solutions in order to minimize the occurrence of these situations.</td>
</tr>
<tr>
<td>History</td>
<td>15/01/2011 Initial approach with architecture definition</td>
</tr>
<tr>
<td></td>
<td>30/05/2011 Released internal version to be reviewed</td>
</tr>
<tr>
<td></td>
<td>02/06/2011 Released final version</td>
</tr>
</tbody>
</table>
Data Flow Diagrams

**FastFix**

- Target application
  - Sends Patches
  - Put data
  - Data Store

**FastFix Core**

- FastFix Core
  - Show Info
  - Actions
  - Get Data

**Admin**

**FastFix Client**

- Context System
  - Context System Information
  - Sends Actions To Context System
  - Apply Patch
  - Report Data

- Client Bridge
  - Sends Actions To Client Self Healing
  - Self Healing Information

- Client Self-Healing
  - Get Patch
  - Put Patch

- Client Data Store
  - Put Context
  - Get Context
# Threat Model: FastFix Architecture

## Threats and Mitigations

### Elements

<table>
<thead>
<tr>
<th>Element Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Flow</td>
<td>Actions</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Apply Patch</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Context System Information</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Event Correlation Interface Usage</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Event Correlation Report</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Fault Replication Interface Usage</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Fault Replication Report</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Get Context</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Get Data</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Get Log Event</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Get Log</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Get Patch Event</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Get Patch</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Patch Generator Interface Usage</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Patch Generator Report</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Provide Information</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Put Context</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Put data</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Put Log Event</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Put Log</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Put Patch Event</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Put Patch</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Report Data</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Report Facts</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Self Healing Information</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Send Controls</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Sends Actions To Client Self Healing</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Sends Actions to Context System</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Sends Patches</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Show Info</td>
</tr>
</tbody>
</table>
## Elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Store</td>
<td>Client Data Store</td>
</tr>
<tr>
<td>Data Store</td>
<td>Data Store</td>
</tr>
<tr>
<td>Data Store</td>
<td>Server DataStore</td>
</tr>
<tr>
<td>Interactor</td>
<td>Admin</td>
</tr>
<tr>
<td>Interactor</td>
<td>FastFix Core</td>
</tr>
<tr>
<td>Interactor</td>
<td>Target application</td>
</tr>
<tr>
<td>Interactor</td>
<td>UI System</td>
</tr>
<tr>
<td>Process</td>
<td>Client Bridge</td>
</tr>
<tr>
<td>Process</td>
<td>Client Self-Healing</td>
</tr>
<tr>
<td>Process</td>
<td>Context System</td>
</tr>
<tr>
<td>Process</td>
<td>Event Correlation</td>
</tr>
<tr>
<td>Process</td>
<td>FastFix Core</td>
</tr>
<tr>
<td>Process</td>
<td>Fault Replication</td>
</tr>
<tr>
<td>Process</td>
<td>Maintenance Bridge</td>
</tr>
<tr>
<td>Process</td>
<td>Patch Generator</td>
</tr>
</tbody>
</table>

## TrustBoundary

- TrustBoundary
- TrustBoundary
- TrustBoundary
- TrustBoundary

## External Interactors

### Threats against Admin

**Spoofing (Threat #185)**

**Threat:** Credentials held on client side should be modified. Possible impacts:

- Pretend to be another user in the system
- Access to not authorized information

**Mitigation:** Credentials should not be stored in client side.

**Spoofing (Threat #261)**

**Threat:** Credentials on server side can be viewed. Possible impacts:

- Retrieve users private information: user and password:
- Access to not authorized information

**Mitigation:** Credentials must be stored in a encrypted form or, at least, with a hash.

**Spoofing (Threat #262)**

**Threat:** User definition could be overridden. Possible impacts:
* Pretend to be another user in the system
* Access to not authorized information

**Mitigation:** User authentication must be unique and has to be confirmed, each login is related to some param that allows unique identification

**Repudiation (Threat #186)**

**Threat:** Information about the actions done by any user should be stored. Possible impacts:

* Impossibility to extract responsibilities on actions
* Claiming not doing something

**Mitigation:** Enable a logging system that, respecting privacy, allow to trace user activity. This log must have a timestamp signature so that it can be identified in time and order of execution.

**Threats against FastFix Core**

**Spoofing (Threat #445)**

**Threat:** Credentials held on client side should be modified. Possible impacts:

* Pretend to be another user in the system
* Access to not authorized information

**Mitigation:** Credentials should not be stored in client side.

**Spoofing (Threat #459)**

**Threat:** Credentials on server side can be viewed. Possible impacts:

* Retrieve users private information: user and password:
* Access to not authorized information

**Mitigation:** Credentials must be stored in a encrypted form or, at least, with a hash.

**Spoofing (Threat #460)**

**Threat:** User definition could be overriden. Possible impacts:

* Pretend to be another user in the system
* Access to not authorized information

**Mitigation:** User authentication must be unique and has to be confirmed, each login is related to some param that allows unique identification

**Repudiation (Threat #446)**

**Threat:** Information about the actions done by any user should be stored. Possible impacts:

* Impossibility to extract responsibilities on actions
* Claiming not doing something

**Mitigation:** Enable a logging system that, respecting privacy, allow to trace user activity. This log must have a timestamp signature so that it can be identified in time and order of execution.

**Threats against Target application**

**Spoofing (Threat #228)**
**Threat:** Credentials held on client side should be modified, Posible impacts:

* Pretend to be another user in the system
* Access to not authorized information

**Mitigation:** Credentials should not be stored in client side.

**Spoofing (Threat #263)**

**Threat:** Credentials on server side can be viewed. Posible impacts:

* Retrieve users private information: user and password:
* Access to not authorized information

**Mitigation:** Credentials must be stored in a encrypted form or, at least, with a hash.

**Spoofing (Threat #264)**

**Threat:** User definition could be overriden. Posible impacts:

* Pretend to be another user in the system
* Access to not authorized information

**Mitigation:** User authentication must be unique and has to be confirmed, each login is related to some param that allows unique identification

**Repudiation (Threat #229)**

**Threat:** Information about the actions done by any user should be stored. Posible impacts:

* Imposibility to extract responsibilities on actions
* Claiming not doing something

**Mitigation:** Enable a logging system that, respecting privacy, allow to trace user activity. this log must have a timestamp signature so that it can be identified in time and order of execution.

**Threats against UI System**

**Spoofing (Threat #455)**

**Threat:** Credentials held on client side should be modified, Posible impacts:

* Pretend to be another user in the system
* Access to not authorized information

**Mitigation:** Credentials should not be stored in client side.

**Spoofing (Threat #461)**

**Threat:** Credentials on server side can be viewed. Posible impacts:

* Retrieve users private information: user and password:
* Access to not authorized information

**Mitigation:** Credentials must be stored in a encrypted form or, at least, with a hash.

**Spoofing (Threat #462)**

**Threat:** User definition could be overriden. Posible impacts:

* Pretend to be another user in the system
* Access to not authorized information

**Mitigation:** User authentication must be unique and has to be confirmed, each login is related to some param that allows unique identification

**Repudiation (Threat #456)**

**Threat:** Information about the actions done by any user should be stored. Possible impacts:

* Imposibility to extract responsibilities on actions
* Claiming not doing something

**Mitigation:** Enable a logging system that, respecting privacy, allow to trace user activity. this log must have a timestamp signature so that it can be identified in time and order of execution.

**Processes**

**Threats against Client Bridge**

**Spoofing (Threat #415)**

**Threat:** Credentials on client side can be viewed. Possible impacts:

* Retrieve users private information: user and password:
* Access to not authorized information

**Mitigation:** Credentials must be stored in a encrypted form or, at least, with a hash.

**Tampering (Threat #416)**

**Threat:** User inputs could include unexpected information. Possible impacts:

* It may crash client application
* It may steal user information

**Mitigation:** User inputs must be validated against a white list in order to limit risks in unknown input data.

**Repudiation (Threat #417)**

**Threat:** Information about the actions done by any user should be stored. Possible impacts:

* Imposibility to extract responsibilities on actions
* Claiming not doing something

**Mitigation:** Enable a logging system that, respecting privacy, allow to trace user activity. this log must have a timestamp signature so that it can be identified in time and order of execution.

**Information Disclosure (Threat #418)**

**Threat:** The information exchanged among the Client Bridge could be stolen. Possible impacts:

* It may steal user information
* It may steal platform information

**Mitigation:** Ensure that systems performs either key exchange or key validation out of band. Client Bridge must consider ACLs and encryption, over the network, data can only be protected by encryption.
**Denial of Service (Threat #419)**

**Threat:** The information exchanged for Client Bridge could be modified. Possible impacts:

* It may crash client application

**Mitigation:** On system: ensure that not everyone can change ACLs on the objects. On a network, ensure that Client Bridge does not use up resources when contacted by an anonymous entity and that enables a timeout mechanism to release inactive connections.

**Elevation of Privilege (Threat #420)**

**Threat:** Inputs could include unexpected information. Possible impacts:

* It may crash client application
* It may steal user information
* It may gain an elevation of privilege

**Mitigation:** All inputs must be validated against a white list in order to limit risks in unknown input data.

**Threats against Client Self-Healing**

**Spoofing (Threat #387)**

**Threat:** Credentials on client side can be viewed. Possible impacts:

* Retrieve users private information: user and password:
* Access to not authorized information

**Mitigation:** Credentials must be stored in a encrypted form or, at least, with a hash.

**Tampering (Threat #388)**

**Threat:** User inputs could include unexpected information. Possible impacts:

* It may crash client application
* It may steal user information

**Mitigation:** User inputs must be validated against a white list in order to limit risks in unknown input data.

**Repudiation (Threat #389)**

**Threat:** Information about the actions done by any user should be stored. Possible impacts:

* Imposibility to extract responsibilities on actions
* Claiming not doing something

**Mitigation:** Enable a logging system that, respecting privacy, allow to trace user activity. This log must have a timestamp signature so that it can be identified in time and order of execution.

**Information Disclosure (Threat #390)**

**Threat:** The information exchanged among the Client Self-Healing System could be stolen. Possible impacts:

* It may steal user information
* It may steal platform information

**Mitigation:** Ensure that systems performs either key exchange or key validation out of band.
Client Self-Healing must consider ACLs and encryption, over the network, data can only be protected by encryption.

**Denial of Service (Threat #391)**

**Threat:** The information exchanged for Client Self-Healing could be modified. Possible impacts:

* It may crash client application

**Mitigation:** On system: ensure that not everyone can change ACLs on the objects. On a network, ensure that Client Self-Healing System does not use up resources when contacted by an anonymous entity and that enables a timeout mechanism to release inactive connections.

**Elevation of Privilege (Threat #392)**

**Threat:** Inputs could include unexpected information. Possible impacts:

* It may crash client application
* It may steal user information
* It may gain an elevation of privilege

**Mitigation:** All inputs must be validated against a white list in order to limit risks in unknown input data.

**Threats against Context System**

**Spoofing (Threat #381)**

**Threat:** Credentials on client side can be viewed. Possible impacts:

* Retrieve users private information: user and password:
  * Access to not authorized information

**Mitigation:** Credentials must be stored in an encrypted form or, at least, with a hash.

**Tampering (Threat #382)**

**Threat:** User inputs could include unexpected information. Possible impacts:

* It may crash client application
* It may steal user information

**Mitigation:** User inputs must be validated against a white list in order to limit risks in unknown input data.

**Repudiation (Threat #383)**

**Threat:** Information about the actions done by any user should be stored. Possible impacts:

* Impossibility to extract responsibilities on actions
* Claiming not doing something

**Mitigation:** Enable a logging system that, respecting privacy, allow to trace user activity. This log must have a timestamp signature so that it can be identified in time and order of execution.

**Information Disclosure (Threat #384)**

**Threat:** The information exchanged among the Context System could be stolen. Possible impacts:

* It may steal user information
* It may steal platform information

**Mitigation:** Ensure that systems performs either key exchange or key validation out of band. Context System must consider ACLs and encryption, over the network, data can only be protected by encryption.

**Denial of Service (Threat #385)**

**Threat:** The information exchanged for Context System could be modified. Possible impacts:

* It may crash client application

**Mitigation:** On system: ensure that not everyone can change ACLs on the objects. On a network, ensure that Context System does not use up resources when contacted by an anonymous entity and that enables a timeout mechanism to release inactive connections

**Elevation of Privilege (Threat #386)**

**Threat:** Inputs could include unexpected information. Possible impacts:

* It may crash client application
* It may steal user information
* It may gain an elevation of privilege

**Mitigation:** All inputs must be validated against a white list in order to limit risks in unknown input data.

**Threats against Event Correlation**

**Spoofing (Threat #274)**

**Threat:** Credentials on client side can be viewed. Possible impacts:

* Retrieve users private information: user and password:
* Access to not authorized information

**Mitigation:** Credentials must be stored in a encrypted form or, at least, with a hash.

**Tampering (Threat #275)**

**Threat:** User inputs could include unexpected information. Possible impacts:

* It may crash client application
* It may steal user information

**Mitigation:** User inputs must be validated against a white list in order to limit risks in unknown input data.

**Repudiation (Threat #276)**

**Threat:** Information about the actions done by any user should be stored. Possible impacts:

* Imposibility to extract responsibilities on actions
* Claiming not doing something

**Mitigation:** Enable a logging system that, respecting privacy, allow to trace user activity. this log must have a timestamp signature so that it can be identified in time and order of execution.

**Information Disclosure (Threat #277)**

**Threat:** The information exchanged among the Event Correlation System could be stolen.
Posible impacts:

* It may steal user information
* It may steal platform information

**Mitigation:** Ensure that systems performs either key exchange or key validation out of band. Event Correlation must consider ACLs and encryption, over the network, data can only be protected by encryption.

**Denial of Service (Threat #278)**

**Threat:** The information exchanged for Event Correlation could be modified. Possible impacts:

* It may crash client application

**Mitigation:** On system: ensure that not everyone can change ACLs on the objects. On a network, ensure that Event Correlation System does not use up resources when contacted by an anonymous entity and that enables a timeout mechanism to release inactive connections.

**Elevation of Privilege (Threat #279)**

**Threat:** Inputs could include unexpected information. Possible impacts:

* It may crash client application
* It may steal user information
* It may gain an elevation of privilege

**Mitigation:** All inputs must be validated against a white list in order to limit risks in unknown input data.

**Threats against FastFix Core**

**Spoofing (Threat #230)**

**Threat:** Credentials on server side can be viewed. Possible impacts:

* Retrieve users private information: user and password:
* Access to not authorized information

**Mitigation:** Credentials must be stored in a encrypted form or, at least, with a hash.

**Tampering (Threat #231)**

**Threat:** User inputs could include unexpected information. Possible impacts:

* It may crash client application
* It may steal user information

**Mitigation:** User inputs must be validated against a white list in order to limit risks in unknown input data.

**Repudiation (Threat #232)**

**Threat:** Information about the actions done by any user should be stored. Possible impacts:

* Imposibility to extract responsibilities on actions
* Claiming not doing something

**Mitigation:** Enable a logging system that, respecting privacy, allow to trace user activity. this log must have a timestamp signature so that it can be identified in time and order of execution.
**Information Disclosure (Threat #233)**

**Threat:** The information exchanged among the FastFix Core could be stolen. Possible impacts:

* It may steal user information
* It may steal platform information

**Mitigation:** Ensure that systems performs either key exchange or key validation out of band. FastFix Core must consider ACLs and encryption, over the network, data can only be protected by encryption.

**Denial of Service (Threat #234)**

**Threat:** The information exchanged for FastFix Core could be modified. Possible impacts:

* It may crash client application

**Mitigation:** On system: ensure that not everyone can change ACLs on the objects. On a network, ensure that FastFix Core does not use up resources when contacted by an anonymous entity and that enables a timeout mechanism to release inactive connections.

**Elevation of Privilege (Threat #235)**

**Threat:** Inputs could include unexpected information. Possible impacts:

* It may crash client application
* It may steal user information
* It may gain an elevation of privilege

**Mitigation:** All inputs must be validated against a white list in order to limit risks in unknown input data.

**Threats against Fault Replication**

**Spoofing (Threat #280)**

**Threat:** Credentials on client side can be viewed. Possible impacts:

* Retrieve users private information: user and password:
* Access to not authorized information

**Mitigation:** Credentials must be stored in an encrypted form or, at least, with a hash.

**Tampering (Threat #281)**

**Threat:** User inputs could include unexpected information. Possible impacts:

* It may crash client application
* It may steal user information

**Mitigation:** User inputs must be validated against a white list in order to limit risks in unknown input data.

**Repudiation (Threat #282)**

**Threat:** Information about the actions done by any user should be stored. Possible impacts:

* Impossibility to extract responsibilities on actions
* Claiming not doing something

**Mitigation:** Enable a logging system that, respecting privacy, allow to trace user activity. This log must have a timestamp signature so that it can be identified in time and order of
**Information Disclosure (Threat #283)**

**Threat:** The information exchanged among the Fault Replication System could be stolen. Possible impacts:

* It may steal user information
* It may steal platform information

**Mitigation:** Ensure that systems performs either key exchange or key validation out of band. Fault Replication must consider ACLs and encryption, over the network, data can only be protected by encryption.

**Denial of Service (Threat #284)**

**Threat:** The information exchanged for Fault Replication could be modified. Possible impacts:

* It may crash client application

**Mitigation:** On system: ensure that not everyone can change ACLs on the objects. On a network, ensure that Fault Replication System does not use up resources when contacted by an anonymous entity and that enables a timeout mechanism to release inactive connections.

**Elevation of Privilege (Threat #285)**

**Threat:** Inputs could include unexpected information. Possible impacts:

* It may crash client application
* It may steal user information
* It may gain an elevation of privilege

**Mitigation:** All inputs must be validated against a white list in order to limit risks in unknown input data.

**Threats against Maintenance Bridge**

**Spoofing (Threat #329)**

**Threat:** Credentials on client side can be viewed. Possible impacts:

* Retrieve users private information: user and password:
* Access to not authorized information

**Mitigation:** Credentials must be stored in a encrypted form or, at least, with a hash.

**Tampering (Threat #330)**

**Threat:** User inputs could include unexpected information. Possible impacts:

* It may crash client application
* It may steal user information

**Mitigation:** User inputs must be validated against a white list in order to limit risks in unknown input data.

**Repudiation (Threat #331)**

**Threat:** Information about the actions done by any user should be stored. Possible impacts:

* Impossibility to extract responsibilities on actions
* Claiming not doing something
**Mitigation:** Enable a logging system that, respecting privacy, allow to trace user activity. this log must have a timestamp signature so that it can be identified in time and order of execution.

**Information Disclosure (Threat #332)**

**Threat:** The information exchanged among the Maintainence Bridge System could be stolen. Possible impacts:

* It may steal user information
* It may steal platform information

**Mitigation:** Ensure that systems performs either key exchange or key validation out of band. Maintenance Bridge must consider ACLs and encryption, over the network, data can only be protected by encryption.

**Denial of Service (Threat #333)**

**Threat:** The information exchanged for Maintenance Bridge could be modified. Possible impacts:

* It may crash client application

**Mitigation:** On system: ensure that not everyone can change ACLs on the objects. On a network, ensure that Maintenance Bridge System does not use up resources when contacted by an anonymous entity and that enables a timeout mechanism to release inactive connections.

**Elevation of Privilege (Threat #334)**

**Threat:** Inputs could include unexpected information. Possible impacts:

* It may crash client application
* It may steal user information
* It may gain an elevation of privilege

**Mitigation:** All inputs must be validated against a white list in order to limit risks in unknown input data.

**Threats against Patch Generator**

**Spoofing (Threat #286)**

**Threat:** Credentials on client side can be viewed. Possible impacts:

* Retrieve users private information: user and password:
* Access to not authorized information

**Mitigation:** Credentials must be stored in a encrypted form or, at least, with a hash.

**Tampering (Threat #287)**

**Threat:** User inputs could include unexpected information. Possible impacts:

* It may crash client application
* It may steal user information

**Mitigation:** User inputs must be validated against a white list in order to limit risks in unknown input data.

**Repudiation (Threat #288)**

**Threat:** Information about the actions done by any user should be stored. Possible impacts:
* Impossibility to extract responsibilities on actions
* Claiming not doing something

**Mitigation:** Enable a logging system that, respecting privacy, allow to trace user activity. This log must have a timestamp signature so that it can be identified in time and order of execution.

**Information Disclosure (Threat #289)**

**Threat:** The information exchanged among the Patch Generator System could be stolen. Possible impacts:

* It may steal user information
* It may steal platform information

**Mitigation:** Ensure that systems perform either key exchange or key validation out of band. Patch Generator must consider ACLs and encryption, over the network, data can only be protected by encryption.

**Denial of Service (Threat #290)**

**Threat:** The information exchanged for Patch Generator could be modified. Possible impacts:

* It may crash client application

**Mitigation:** On system: ensure that not everyone can change ACLs on the objects. On a network, ensure that Patch Generator System does not use up resources when contacted by an anonymous entity and that enables a timeout mechanism to re-release inactive connections.

**Elevation of Privilege (Threat #291)**

**Threat:** Inputs could include unexpected information. Possible impacts:

* It may crash client application
* It may steal user information
* It may gain an elevation of privilege

**Mitigation:** All inputs must be validated against a white list in order to limit risks in unknown input data.

**Data Flows**

**Threats against Actions**

**Tampering (Threat #248)**

**Threat:** The information exchanged among the Admin Application and FastFix Core could be modified. Possible impacts:

* It may crash server application
* It may steal user information
* It may apply wrong patches

**Mitigation:** Either cryptographic integrity controls or good use of the OS features to protect communications. FastFix Core needs to protect dataflows implementing timestamped/sequenced and integrity checks. Use an anti-replay techniques (investigate sequence numbers before timers). Also it is necessary to check the dataflows for duplicate/overlapped data.
FastFix Core and target application must authenticate each other with keys obtained or validated out of band

**Information Disclosure (Threat #249)**

**Threat:** The information exchanged among the Admin Application and FastFix Core could be modified. Possible impacts:

* It may crash server application
* It may steal user information
* It may apply wrong patches

**Mitigation:** Ensure that systems performs either key exchange or key validation out of band. FastFix Core must consider ACLs and encryption, over the network, data can only be protected by encryption.

**Denial of Service (Threat #250)**

**Threat:** The information exchanged among the FastFix Core and the Admin Application could be modified. Possible impacts:

* It may crash server application

**Mitigation:** On system: ensure that not everyone can change ACLs on the objects. On a network, ensure that FastFix Core does not use up resources when contacted by an anonymous entity

**Threats against Apply Patch**

**Tampering (Threat #424)**

**Threat:** The information exchanged among the FastFix Core and the Client Bridge could be modified. Possible impacts:

* It may crash client application
* It may steal user information
* It may apply wrong patches

**Mitigation:** Either cryptographic integrity controls or good use of the OS features to protect communications. FastFix Core needs to protect dataflows implementing timestamped/sequenced and integrity checks. Use an anti-replay techniques (investigate sequence numbers before timers). Also it is necessary to check the dataflows for duplicate/overlapped data. FastFix Core and Client Bridge must authenticate each other with keys obtained or validated out of band.

**Information Disclosure (Threat #425)**

**Threat:** The information exchanged among the FastFix Core and the Client Bridge could be stolen. Possible impacts:

* It may steal user information
* It may steal platform information

**Mitigation:** Ensure that systems performs either key exchange or key validation out of band. FastFix Core must consider ACLs and encryption, over the network, data can only be protected by encryption.

**Denial of Service (Threat #426)**
**Threat:** The information exchanged among the FastFix Core and the Client Bridge could be modified. Possible impacts:

* It may crash client application

**Mitigation:** On system: ensure that not everyone can change ACLs on the objects. On a network, ensure that FastFix Core does not use up resources when contacted by an anonymous entity.

### Threats against Provide Information

#### Tampering (Threat #375)

**Threat:** The information exchanged among the FastFix Core and the UI System could be modified. Possible impacts:

* It may crash client application
* It may steal user information
* It may apply wrong patches

**Mitigation:** Either cryptographic integrity controls or good use of the OS features to protect communications. FastFix Core needs to protect dataflows implementing timestamped/sequenced and integrity checks. Use an anti-replay techniques (investigate sequence numbers before timers). Also it is necessary to check the dataflows for duplicate/overlapped data.

FastFix Core and target application must authenticate each other with keys obtained or validated out of band.

#### Information Disclosure (Threat #376)

**Threat:** The information exchanged among the FastFix Core and UI System could be stolen. Possible impacts:

* It may steal user information
* It may steal platform information

**Mitigation:** Ensure that systems performs either key exchange or key validation out of band. FastFix Core must consider ACLs and encryption, over the network, data can only be protected by encryption.

#### Denial of Service (Threat #377)

**Threat:** The information exchanged among the FastFix Core and the UI System could be modified. Possible impacts:

* It may crash client application

**Mitigation:** On system: ensure that not everyone can change ACLs on the objects. On a network, ensure that FastFix Core does not use up resources when contacted by an anonymous entity.

### Threats against Report Data

#### Tampering (Threat #421)

**Threat:** The information exchanged among the Client Bridge and FastFix Core could be modified. Possible impacts:

* It may crash server application
* It may steal user information
* It may apply wrong patches

**Mitigation:** Either cryptographic integrity controls or good use of the OS features to protect communications. FastFix Core needs to protect dataflows implementing timestamped/sequenced and integrity checks. Use an anti-replay techniques (investigate sequence numbers before timers). Also it is necessary to check the dataflows for duplicate/overlapped data.

FastFix Core and Client Bridge must authenticate each other with keys obtained or validated out of band.

**Information Disclosure (Threat #422)**

**Threat:** The information exchanged among the Client Bridge and FastFix Core could be modified. Possible impacts:

* It may crash server application
* It may steal user information
* It may apply wrong patches

**Mitigation:** Ensure that systems performs either key exchange or key validation out of band. FastFix Core must consider ACLs and encryption, over the network, data can only be protected by encryption.

**Denial of Service (Threat #423)**

**Threat:** The information exchanged among the FastFix Core and the Client Bridge could be modified. Possible impacts:

* It may crash server application

**Mitigation:** On system: ensure that not everyone can change ACLs on the objects. On a network, ensure that FastFix Core does not use up resources when contacted by an anonymous entity.

**Threats against Report Facts**

**Tampering (Threat #245)**

**Threat:** The information exchanged among the Target Application and FastFix Core could be modified. Possible impacts:

* It may crash server application
* It may steal user information
* It may apply wrong patches

**Mitigation:** Either cryptographic integrity controls or good use of the OS features to protect communications. FastFix Core needs to protect dataflows implementing timestamped/sequenced and integrity checks. Use an anti-replay techniques (investigate sequence numbers before timers). Also it is necessary to check the dataflows for duplicate/overlapped data.

FastFix Core and target application must authenticate each other with keys obtained or validated out of band.

**Information Disclosure (Threat #246)**

**Threat:** The information exchanged among the Target Application and FastFix Core could be modified. Possible impacts:

* It may crash server application
* It may steal user information
* It may apply wrong patches

**Mitigation:** Ensure that systems performs either key exchange or key validation out of band. FastFix Core must consider ACLs and encryption, over the network, data can only be protected by encryption.

**Denial of Service (Threat #247)**

**Threat:** The information exchanged among the FastFix Core and the Target Application could be modified. Possible impacts:

* It may crash server application

**Mitigation:** On system: ensure that not everyone can change ACLs on the objects. On a network, ensure that FastFix Core does not use up resources when contacted by an anonymous entity.

**Threats against Send Controls**

**Tampering (Threat #378)**

**Threat:** The information exchanged among the UI System and Maintenance Bridge could be modified. Possible impacts:

* It may crash server application
* It may steal user information
* It may apply wrong patches

**Mitigation:** Either cryptographic integrity controls or good use of the OS features to protect communications. Maintenance Bridge needs to protect dataflows implementing timestamped/sequenced and integrity checks. Use an anti-replay techniques (investigate sequence numbers before timers). Also it is necessary to check the dataflows for duplicate/overlapped data.

UI System and Maintenance Bridge must authenticate each other with keys obtained or validated out of band.

**Information Disclosure (Threat #379)**

**Threat:** The information exchanged among the Maintenance Bridge and UI System could be modified. Possible impacts:

* It may crash server application
* It may steal user information
* It may apply wrong patches

**Mitigation:** Ensure that systems performs either key exchange or key validation out of band. Maintenance Bridge must consider ACLs and encryption, over the network, data can only be protected by encryption.

**Denial of Service (Threat #380)**

**Threat:** The information exchanged among the UI System and the Maintenance Bridge could be modified. Possible impacts:

* It may crash server application

**Mitigation:** On system: ensure that not everyone can change ACLs on the objects. On a network, ensure that Maintenance Server does not use up resources when contacted by an anonymous entity.
Threats against Sends Patches

**Tampering (Threat #242)**

**Threat:** The information exchanged among the FastFix Core and the Target Application could be modified. Possible impacts:

* It may crash client application
* It may steal user information
* It may apply wrong patches

**Mitigation:** Either cryptographic integrity controls or good use of the OS features to protect communications. FastFix Core needs to protect dataflows implementing timestamped/sequenced and integrity checks. Use an anti-replay techniques (investigate sequence numbers before timers). Also it is necessary to check the dataflows for duplicate/overlapped data. FastFix Core and target application must authenticate each other with keys obtained or validated out of band

**Information Disclosure (Threat #243)**

**Threat:** The information exchanged among the FastFix Core and the Target Application could be stolen. Possible impacts:

* It may steal user information
* It may steal platform information

**Mitigation:** Ensure that systems performs either key exchange or key validation out of band. FastFix Core must consider ACLs and encryption, over the network, data can only be protected by encryption.

**Denial of Service (Threat #244)**

**Threat:** The information exchanged among the FastFix Core and the Target Application could be modified. Possible impacts:

* It may crash client application

**Mitigation:** On system: ensure that not everyone can change ACLs on the objects. On a network, ensure that FastFix Core does not use up resources when contacted by an anonymous entity.

Threats against Show Info

**Tampering (Threat #257)**

**Threat:** The information exchanged among the FastFix Core and the Target Application could be modified. Possible impacts:

* It may crash client application
* It may steal user information
* It may apply wrong patches

**Mitigation:** Either cryptographic integrity controls or good use of the OS features to protect communications. FastFix Core needs to protect dataflows implementing timestamped/sequenced and integrity checks. Use an anti-replay techniques (investigate sequence numbers before timers). Also it is necessary to check the dataflows for duplicate/overlapped data. FastFix Core and target application must authenticate each other with keys obtained or validated out of band
**Information Disclosure (Threat #258)**

**Threat:** The information exchanged among the FastFix Core and the Target Application could be stolen. Possible impacts:

* It may steal user information
* It may steal platform information

**Mitigation:** Ensure that systems performs either key exchange or key validation out of band. FastFix Core must consider ACLs and encryption, over the network, data can only be protected by encryption.

**Denial of Service (Threat #259)**

**Threat:** The information exchanged among the FastFix Core and the Target Application could be modified. Possible impacts:

* It may crash client application

**Mitigation:** On system: ensure that not everyone can change ACLs on the objects. On a network, ensure that FastFix Core does not use up resources when contacted by an anonymous entity.

**Data Stores**

**Threats against Client Data Store**

**Tampering (Threat #393)**

**Threat:** Data store information can be retrieved or modified if access to the filesystem is granted. Possible impacts:

* It may export all data to untrusted data store
* It may access to private information

**Mitigation:** Either cryptographic integrity controls or good use of the OS features to protect access to the data store

**Repudiation (Threat #394)**

**Threat:** Information stored in the Data Store should be accessed in order to modify its information. Possible impacts:

* Modification on stored facts
* Modification on success rate on patches applicacion.

**Mitigation:** Ensure that access is granted only to defined users and with restricted permission in order to edit / modify / erase contents in the data Store. Make periodic audits of the contents and comparison between backup copies and data in production environment

**Information Disclosure (Threat #395)**

**Threat:** Access to the data store can be configured with different roles and permissions. Possible impacts:

* Access to unauthorized information
* Modifications on model

**Mitigation:** Either cryptographic integrity controls or good use of the OS features to protect
integrity.
Define different users for management and user queries

**Denial of Service (Threat #396)**

**Threat:** Information stored in the data store can increase continuously due to new information sent in order to be stored. Possible impacts:

* System not accessible to retrieve / put information

**Mitigation:** On system: ensure that space available is monitored and force a notification when level raises a defined threshold

**Denial of Service (Threat #458)**

**Threat:** The Data Store has a defined number of connections that can access in order to gather information that can increase uncontrolled. Possible impacts:

* System not accessible to retrieve / put information

**Mitigation:** On system: ensure that connections are monitored and force a notification when level raises a defined threshold

**Threats against Data Store**

**Tampering (Threat #191)**

**Threat:** Data store information can be retrieved or modified if access to the filesystem is granted. Possible impacts:

* It may export all data to untrusted data store
* It may access to private information

**Mitigation:** Either cryptographic integrity controls or good use of the OS features to protect access to the data store

**Repudiation (Threat #192)**

**Threat:** Information stored in the Data Store should be accessed in order to modify its information. Possible impacts:

* Modification on stored facts
* Modification on success rate on patches application.

**Mitigation:** Ensure that access is granted only to defined users and with restricted permission in order to edit / modify / erase contents in the data store. Make periodic audits of the contents and comparison between backup copies and data in production environment

**Information Disclosure (Threat #193)**

**Threat:** Access to the data store can be configured with different roles and permissions. Possible impacts:

* Access to unauthorized information
* Modifications on model

**Mitigation:** Either cryptographic integrity controls or good use of the OS features to protect integrity.
Define different users for management and user queries
<table>
<thead>
<tr>
<th>Threat against Server DataStore</th>
<th>Threat #194: Denial of Service</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Threat:</strong></td>
<td>Information stored in the data store can increase continuously due to new information sent in order to be stored. Possible impacts:</td>
</tr>
<tr>
<td>*</td>
<td>System not accessible to retrieve / put information</td>
</tr>
<tr>
<td><strong>Mitigation:</strong></td>
<td>On system: ensure that space available is monitored and force a notification when level raises a defined threshold</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Threat #260: Denial of Service</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Threat:</strong></td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td><strong>Mitigation:</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Threats against Server DataStore</th>
<th>Threat #335: Tampering</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Threat:</strong></td>
<td>Data store information can be retrieved or modified if access to the filesystem is granted. Possible impacts:</td>
</tr>
<tr>
<td>*</td>
<td>It may export all data to untrusted data store</td>
</tr>
<tr>
<td>*</td>
<td>It may access to private information</td>
</tr>
<tr>
<td><strong>Mitigation:</strong></td>
<td>Either cryptographic integrity controls or good use of the OS features to protect access to the data store</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Threat #336: Repudiation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Threat:</strong></td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td><strong>Mitigation:</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Threat #337: Information Disclosure</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Threat:</strong></td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td><strong>Mitigation:</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Threat #338: Denial of Service</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Threat:</strong></td>
</tr>
</tbody>
</table>
sent in order to be stored. Possible impacts:

* System not accessible to retrieve / put information

**Mitigation:** On system: ensure that space available is monitored and force a notification when level raises a defined threshold

**Denial of Service (Threat #457)**

**Threat:** The Data Store has a defined number of connections that can access in order to gather information that can increase uncontrolled. Possible impacts:

* System not accessible to retrieve / put information

**Mitigation:** On system: ensure that connections are monitored and force a notification when level raises a defined threshold
## Threat Model: FastFix Architecture

### Certifications

<table>
<thead>
<tr>
<th>User Alias</th>
<th>Element Name</th>
<th>Threat Type</th>
<th>Reason for certification</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>David</td>
<td>Context System</td>
<td>Tampering</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Client Bridge and Context System is done in the limits of a trust boundary (same machine). Only the user has access or can tamper the data exchanged so, Tampering threats do not apply.</td>
</tr>
<tr>
<td>Monteagudo</td>
<td>Information</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>David</td>
<td>Context System</td>
<td>InformationDisclosure</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Client Bridge and Context System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Information Disclosure threats do not apply.</td>
</tr>
<tr>
<td>Monteagudo</td>
<td>Information</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>David</td>
<td>Context System</td>
<td>DenialOfService</td>
<td>within a trust boundary</td>
<td>All the information exchanged among the Client Bridge and the Context System is done in the limits of a trust boundary (same machine). It is mandatory, not to use TCP/UDP sockets binded to * (comunications among them must be set up within a trust or process boundary ). It means that there is no possibility to connect from outside the Client Bridge to the Context System services/API provided.</td>
</tr>
<tr>
<td>Monteagudo</td>
<td>Information</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>David</td>
<td>Event Correlation</td>
<td>Tampering</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Maintenance Bridge and Event Correlation System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Tampering threats do not apply.</td>
</tr>
<tr>
<td>Monteagudo</td>
<td>Interface Usage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>David</td>
<td>Event Correlation</td>
<td>InformationDisclosure</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Maintenance Bridge and Event Correlation System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Information Disclosure threats do not apply.</td>
</tr>
<tr>
<td>Monteagudo</td>
<td>Interface Usage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>David</td>
<td>Event Correlation</td>
<td>DenialOfService</td>
<td>within a trust boundary</td>
<td>All the information exchanged among the Maintenance Bridge and the Event Correlation System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, DenialOfService threats do not apply.</td>
</tr>
<tr>
<td>Monteagudo</td>
<td>Interface Usage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Alias</td>
<td>Element Name</td>
<td>Threat Type</td>
<td>Reason for certification</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------</td>
<td>-------------------</td>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Usage</td>
<td>Tampering</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Maintenance Bridge and Event Correlation System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Tampering threats do not apply.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Event Correlation Report</td>
<td>InformationDisclosure</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Data Store and Event Correlation System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Information Disclosure threats do not apply.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Event Correlation Report</td>
<td>DenialOfService</td>
<td>within a trust boundary</td>
<td>All the information exchanged among the Maintenance Bridge and the Event Correlation System is done in the limits of a trust boundary (same machine). It is mandatory, not to use TCP/UDP sockets binded to * (communications among them must be set up within a trust or process boundary ). It means that there is no possibility to connect from outside the Maintenance Bridge to the Event Correlation System services/API provided.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Fault Replication Interface Usage</td>
<td>Tampering</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Maintenance Bridge and Fault Replication System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Tampering threats do not apply.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Fault Replication Interface Usage</td>
<td>InformationDisclosure</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Maintenance Bridge and Fault Replication System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Information Disclosure threats do not apply.</td>
</tr>
<tr>
<td>User Alias</td>
<td>Element Name</td>
<td>Threat Type</td>
<td>Reason for certification</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------</td>
<td>------------------</td>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Fault Replication Interface Usage</td>
<td>DenialOfService</td>
<td>within a trust boundary</td>
<td>All the information exchanged among the Maintenance Bridge and the Fault Replication System is done in the limits of a trust boundary (same machine). It is mandatory, not to use TCP/UDP sockets binded to * (comunications among them must be set up within a trust or process boundary ). It means that there is no possibility to connect from outside the Maintenance Bridge to the Fault Replication System services/API provided.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Fault Replication Report</td>
<td>Tampering</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Maintenance Bridge and Fault Replication System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Tampering threats do not apply.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Fault Replication Report</td>
<td>InformationDisclosure</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Maintenance Bridge and Fault Replication System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Information Disclosure threats do not apply.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Fault Replication Report</td>
<td>DenialOfService</td>
<td>within a trust boundary</td>
<td>All the information exchanged among the Maintenance Bridge and the Fault Replication System is done in the limits of a trust boundary (same machine). It is mandatory, not to use TCP/UDP sockets binded to * (comunications among them must be set up within a trust or process boundary ). It means that there is no possibility to connect from outside the Maintenance Bridge to the Fault Replication System services/API provided.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Get Context Tampering</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Data Store and Context System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Tampering threats do not apply.</td>
<td></td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>User Alias</th>
<th>Element Name</th>
<th>Threat Type</th>
<th>Reason for certification</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>David Monteagudo</td>
<td>Get Context</td>
<td>Information Disclosure</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Data Store and Context System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Information Disclosure threats do not apply.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Get Context</td>
<td>DenialOfService</td>
<td>within a trust boundary</td>
<td>All the information exchanged among the Data Store and the Context System is done in the limits of a trust boundary (same machine). It is mandatory, not to use TCP/UDP sockets binded to * (comunications among them must be set up within a trust or process boundary ). It means that there is no possibility to connect from outside the FastFix core to the Data Store services/API provided.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Get Data</td>
<td>Tampering</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Data Store and FastFix Core is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Tampering threats do not apply.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Get Data</td>
<td>Information Disclosure</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Data Store and FastFix Core is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Information Disclosure threats do not apply.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Get Data</td>
<td>DenialOfService</td>
<td>within a trust boundary</td>
<td>All the information exchanged among the Data Store and the FastFix Core is done in the limits of a trust boundary (same machine). It is mandatory, not to use TCP/UDP sockets binded to * (comunications among them must be set up within a trust or process boundary ). It means that there is no possibility to connect from outside the FastFix core to the Data Store services/API provided.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Get Log Event</td>
<td>Tampering</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Data Store and Event Correlation System is done in the limits of a trust boundary (same machine). Only the administrators has access or</td>
</tr>
<tr>
<td>User Alias</td>
<td>Element Name</td>
<td>Threat Type</td>
<td>Reason for certification</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>----------------------</td>
<td>--------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Get Log Event</td>
<td>InformationDisclosure within a trust boundary</td>
<td></td>
<td>All the information exchanged among Data Store and the Event Correlation System is done in the limits of a trust boundary (same machine). Only the administrators have access or can tamper, so, Information Disclosure threats do not apply.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Get Log Event</td>
<td>DenialOfService within a trust boundary</td>
<td></td>
<td>All the information exchanged among the Data Store and the Event Correlation is done in the limits of a trust boundary (same machine). It is mandatory not to use TCP/UDP sockets binded to * (communications among them must be set up within a trust or process boundary). It means that there is no possibility to connect from outside the FastFix core to the Data Store services/API provided.                                                                                                                                                                                                %%%%</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Get Log Event</td>
<td>Tampering within a trust boundary</td>
<td></td>
<td>All the information exchanged among Data Store and Fault Replication is done in the limits of a trust boundary (same machine). Only the administrators have access or can tamper, so, Tampering threats do not apply.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Get Log Event</td>
<td>InformationDisclosure within a trust boundary</td>
<td></td>
<td>All the information exchanged among Data Store and Fault Replication is done in the limits of a trust boundary (same machine). Only the administrators have access or can tamper, so, Information Disclosure threats do not apply.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Get Log Event</td>
<td>DenialOfService within a trust boundary</td>
<td></td>
<td>All the information exchanged among the Data Store and the Fault Replication is done in the limits of a trust boundary (same machine). It is mandatory, not to use TCP/UDP sockets binded to * (communications among them must be set up within a trust or process boundary). It means that there is no possibility to connect from outside the FastFix core to the Data Store services/API provided.</td>
</tr>
<tr>
<td>User Alias</td>
<td>Element Name</td>
<td>Threat Type</td>
<td>Reason for certification</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-----------------</td>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>David</td>
<td>Monteagudo</td>
<td>Get Patch</td>
<td>Tampering</td>
<td>within a trust boundary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Event</td>
<td></td>
<td>All the information exchanged among Data Store and Patch Generator System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Tampering threats do not apply.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Get Patch</td>
<td>InformationDisclosure</td>
<td>within a trust boundary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Event</td>
<td></td>
<td>All the information exchanged among Data Store and Patch Generator System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Information Disclosure threats do not apply.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Get Patch</td>
<td>DenialOfService</td>
<td>within a trust boundary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Event</td>
<td></td>
<td>All the information exchanged among the Data Store and the Patch Generator System is done in the limits of a trust boundary (same machine). It is mandatory, not to use TCP/UDP sockets binded to * (comunications among them must be set up within a trust or process boundary ). It means that there is no possibility to connect from outside the FastFix core to the Data Store services/API provided.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Get Patch</td>
<td>Tampering</td>
<td>within a trust boundary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Event</td>
<td></td>
<td>All the information exchanged among Data Store and Client Self-Healing is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Tampering threats do not apply.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Get Patch</td>
<td>InformationDisclosure</td>
<td>within a trust boundary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Event</td>
<td></td>
<td>All the information exchanged among Data Store and Client Self-Healing is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Information Disclosure threats do not apply.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Get Patch</td>
<td>DenialOfService</td>
<td>within a trust boundary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Event</td>
<td></td>
<td>All the information exchanged among the Data Store and the Client Self-Healing system is done in the limits of a trust boundary (same machine). It is mandatory, not to use TCP/UDP sockets binded to * (comunications among them must be set up within a trust or process boundary ). It means that there is no possibility to connect from outside the FastFix core to the Data Store services/API provided.</td>
</tr>
<tr>
<td>User Alias</td>
<td>Element Name</td>
<td>Threat Type</td>
<td>Reason for certification</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------</td>
<td>----------------------</td>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Patch Generator Interface</td>
<td>Tampering</td>
<td>within a trust boundary</td>
<td>possibility to connect from outside the FastFix core to the Data Store</td>
</tr>
<tr>
<td></td>
<td>Usage</td>
<td></td>
<td></td>
<td>services/API provided.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Patch Generator Interface</td>
<td>InformationDisclosure</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Maintenance Bridge and Patch Generator System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Information Disclosure threats do not apply.</td>
</tr>
<tr>
<td></td>
<td>Usage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Patch Generator Interface</td>
<td>DenialOfService</td>
<td>within a trust boundary</td>
<td>All the information exchanged among the Maintenance Bridge and the Patch Generator System is done in the limits of a trust boundary (same machine). It is mandatory, not to use TCP/UDP sockets binded to * (comunications among them must be set up within a trust or process boundary ). It means that there is no possibility to connect from outside the Maintenance Bridge to the Patch Generator System services/API provided.</td>
</tr>
<tr>
<td></td>
<td>Usage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Patch Generator Report</td>
<td>Tampering</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Maintenance Bridge and Patch Generator System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Tampering threats do not apply.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Patch Generator Report</td>
<td>InformationDisclosure</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Maintenance Bridge and Patch Generator System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Information Disclosure threats do not apply.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Patch Generator Report</td>
<td>DenialOfService</td>
<td>within a trust boundary</td>
<td>All the information exchanged among the Maintenance Bridge and the Patch Generator System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, DenialOfService threats do not apply.</td>
</tr>
<tr>
<td>User Alias</td>
<td>Element Name</td>
<td>Threat Type</td>
<td>Reason for certification</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------</td>
<td>-------------------</td>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Put Context Tampering within a trust boundary</td>
<td>All the information exchanged among Data Store and FastFix Client are done in the limits of a trust boundary (same machine). Only the user has access or can tamper the data exchanged so, Tampering threats do not apply.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Put Context InformationDisclosure within a trust boundary</td>
<td>All the information exchanged among Data Store and Context System are done in the limits of a trust boundary (same machine). Only the user has access or can tamper the data exchanged so, Information Disclosure threats do not apply.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Put Context DenialOfService within a trust boundary</td>
<td>All the information exchanged among the Data Store and the Context System is done in the limits of a trust boundary (same machine). It is mandatory, not to use TCP/UDP sockets binded to * (comunications among them must be set up within a trust or process boundary ). It means that there is no possibility to connect from outside the FastFix Client to the Data Store services/API provided.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Put data Tampering within a trust boundary</td>
<td>All the information exchanged among Data Store and FastFix Core are done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Tampering threats do not apply.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Put data InformationDisclosure within a trust boundary</td>
<td>All the information exchanged among Data Store and FastFix Core are done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Information Disclosure threats do not apply.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Alias</td>
<td>Element Name</td>
<td>Threat Type</td>
<td>Reason for certification</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------</td>
<td>---------------------</td>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>David</td>
<td>Put data</td>
<td>DenialOfService</td>
<td>within a trust boundary</td>
<td>All the information exchanged among the Data Store and the FastFix Core is done in the limits of a trust boundary (same machine). It is mandatory, not to use TCP/UDP sockets binded to * (communications among them must be set up within a trust or process boundary). It means that there is no possibility to connect from outside the FastFix core to the Data Store services/API provided.</td>
</tr>
<tr>
<td>Monteagudo</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>David</td>
<td>Put Log Event</td>
<td>Tampering</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Data Store and Event Correlation is done in the limits of a trust boundary (same machine). Only the user has access or can tamper the data exchanged so, Tampering threats do not apply.</td>
</tr>
<tr>
<td>Monteagudo</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>David</td>
<td>Put Log Event</td>
<td>InformationDisclosure</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Data Store and Event Correlation is done in the limits of a trust boundary (same machine). Only the user has access or can tamper the data exchanged so, Information Disclosure threats do not apply.</td>
</tr>
<tr>
<td>Monteagudo</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>David</td>
<td>Put Log Event</td>
<td>DenialOfService</td>
<td>within a trust boundary</td>
<td>All the information exchanged among the Data Store and the Event Correlation System is done in the limits of a trust boundary (same machine). It is mandatory, not to use TCP/UDP sockets binded to * (communications among them must be set up within a trust or process boundary). It means that there is no possibility to connect from outside the FastFix core to the Data Store services/API provided.</td>
</tr>
<tr>
<td>Monteagudo</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>David</td>
<td>Put Log Event</td>
<td>Tampering</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Data Store and Fault Replication is done in the limits of a trust boundary (same machine). Only the user has access or can tamper the data exchanged so, Tampering threats do not apply.</td>
</tr>
<tr>
<td>Monteagudo</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>David</td>
<td>Put Log Event</td>
<td>InformationDisclosure</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Data Store and Fault Replication is done in the limits of a trust boundary (same machine). Only the user has access or can tamper the data exchanged so, Information Disclosure threats do not apply.</td>
</tr>
<tr>
<td>Monteagudo</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Alias</td>
<td>Element Name</td>
<td>Threat Type</td>
<td>Reason for certification</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>---------------------</td>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Put Log</td>
<td>DenialOfService</td>
<td>within a trust boundary</td>
<td>All the information exchanged among the Data Store and the Fault Replication System is done in the limits of a trust boundary (same machine). It is mandatory, not to use TCP/UDP sockets binded to * (communications among them must be set up within a trust or process boundary). It means that there is no possibility to connect from outside the FastFix core to the Data Store services/API provided.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Put Patch Event</td>
<td>Tampering</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Data Store and Patch Generator is done in the limits of a trust boundary (same machine). Only the user has access or can tamper the data exchanged so, Tampering threats do not apply.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Put Patch Event</td>
<td>InformationDisclosure</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Data Store and Patch Generator is done in the limits of a trust boundary (same machine). Only the user has access or can tamper the data exchanged so, Information Disclosure threats do not apply.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Put Patch Event</td>
<td>DenialOfService</td>
<td>within a trust boundary</td>
<td>All the information exchanged among the Data Store and the Patch Generator System is done in the limits of a trust boundary (same machine). It is mandatory, not to use TCP/UDP sockets binded to * (communications among them must be set up within a trust or process boundary). It means that there is no possibility to connect from outside the FastFix core to the Data Store services/API provided.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Put Patch Event</td>
<td>Tampering</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Data Store and FastFix Client is done in the limits of a trust boundary (same machine). Only the user has access or can tamper the data exchanged so, Tampering threats do not apply.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Put Patch Event</td>
<td>InformationDisclosure</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Data Store and Client Self-Healin are done in the limits of a trust boundary</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User Alias</th>
<th>Element Name</th>
<th>Threat Type</th>
<th>Reason for certification</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>David Monteagudo</td>
<td>Put Patch</td>
<td>DenialOfService</td>
<td>within a trust boundary</td>
<td>All the information exchanged among the Data Store and the Client Self Healing System is done in the limits of a trust boundary (same machine). It is mandatory, not to use TCP/UDP sockets binded to * (communications among them must be set up within a trust or process boundary). It means that there is no possibility to connect from outside the FastFix Client to the Data Store services/API provided.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Self Healing Information</td>
<td>Tampering</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Client Bridge and Client Self-Healing System is done in the limits of a trust boundary (same machine). Only the user has access or can tamper the data exchanged so, Tampering threats do not apply.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Self Healing Information</td>
<td>InformationDisclosure</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Client Bridge and Client Self-Healing System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Information Disclosure threats do not apply.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Self Healing Information</td>
<td>DenialOfService</td>
<td>within a trust boundary</td>
<td>All the information exchanged among the Client Bridge and the Client Self-Healing System is done in the limits of a trust boundary (same machine). It is mandatory, not to use TCP/UDP sockets binded to * (communications among them must be set up within a trust or process boundary). It means that there is no possibility to connect from outside the Client Bridge to the Client Self-Healing System services/API provided.</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Sends Actions To Client Self Healing</td>
<td>Tampering</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Client Bridge and Client Self-Healing System is done in the limits of a trust boundary (same machine). Only the user has access or can tamper the data exchanged so, Tampering threats do not apply.</td>
</tr>
<tr>
<td>User Alias</td>
<td>Element Name</td>
<td>Threat Type</td>
<td>Reason for certification</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------</td>
<td>---------------------</td>
<td>--------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>David Monteagudo</td>
<td>Sends Actions To Client Self Healing</td>
<td>InformationDisclosure</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Client Bridge and Client Self-Healing System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Information Disclosure threats do not apply.</td>
</tr>
<tr>
<td></td>
<td>Sends Actions To Client Self Healing</td>
<td>DenialOfService</td>
<td>within a trust boundary</td>
<td>All the information exchanged among the Client Bridge and the Client Self-Healing System is done in the limits of a trust boundary (same machine). It is mandatory, not to use TCP/UDP sockets binded to * (comunications among them must be set up within a trust or process boundary ). It means that there is no possibility to connect from outside the Client Bridge to the Client Self-Healing System services/API provided.</td>
</tr>
<tr>
<td></td>
<td>Sends Actions to Context System</td>
<td>Tampering</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Client Bridge and Context System is done in the limits of a trust boundary (same machine). Only the user has access or can tamper the data exchanged so, Tampering threats do not apply.</td>
</tr>
<tr>
<td></td>
<td>Sends Actions to Context System</td>
<td>InformationDisclosure</td>
<td>within a trust boundary</td>
<td>All the information exchanged among Client Bridge and Context System is done in the limits of a trust boundary (same machine). Only the administrators has access or can tamper the data exchanged so, Information Disclosure threats do not apply.</td>
</tr>
<tr>
<td></td>
<td>Sends Actions to Context System</td>
<td>DenialOfService</td>
<td>within a trust boundary</td>
<td>All the information exchanged among the Client Bridge and the Context System is done in the limits of a trust boundary (same machine). It is mandatory, not to use TCP/UDP sockets binded to * (comunications among them must be set up within a trust or process boundary ). It means that there is no possibility to connect from outside the Client Bridge to the Context System services/API provided.</td>
</tr>
</tbody>
</table>
Threat Model: FastFix Architecture

External Dependencies

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>URL</th>
<th>Origin</th>
<th>Team Owner</th>
<th>External Owner</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SVN</td>
<td>Not Available</td>
<td>External</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>TRAC</td>
<td>Not Available</td>
<td>External</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Implementation Assumptions

<table>
<thead>
<tr>
<th>ID</th>
<th>Date/Time</th>
<th>Element Impacted</th>
<th>Assumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>04/27/2011 10:02:00AM</td>
<td>Event Correlation</td>
<td>All communications will be between components will take place using Server Bridge</td>
</tr>
<tr>
<td>2</td>
<td>04/27/2011 10:02:00AM</td>
<td>Fault Replication</td>
<td>All communications will be between components will take place using Server Bridge</td>
</tr>
<tr>
<td>3</td>
<td>04/27/2011 10:02:00AM</td>
<td>Patch Generator</td>
<td>All communications will be between components will take place using Server Bridge</td>
</tr>
<tr>
<td>4</td>
<td>04/27/2011 10:02:00AM</td>
<td>Maintenance Bridge</td>
<td>All communications will be between components will take place using Server Bridge</td>
</tr>
<tr>
<td>5</td>
<td>04/27/2011 10:02:00AM</td>
<td>Context System</td>
<td>All communications will be between components will take place using Client Bridge</td>
</tr>
<tr>
<td>6</td>
<td>04/27/2011 10:02:00AM</td>
<td>Client Self-Healing</td>
<td>All communications will be between components will take place using Client Bridge</td>
</tr>
<tr>
<td>7</td>
<td>04/27/2011 10:02:00AM</td>
<td>Client Bridge</td>
<td>All communications will be between components will take place using Client Bridge</td>
</tr>
</tbody>
</table>

External Security Notes

<table>
<thead>
<tr>
<th>ID</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Usage of OWASP libraries must be considered in order to develop an application in a secure way</td>
</tr>
</tbody>
</table>