# Threat Modeling Report

Created on 6/21/2015 6:47:08 PM

Threat Model Name:

Owner:

Reviewer:

Contributors:

Description:

Assumptions:

External Dependencies:

## Threat Model Summary:

| | |
|---|---|
| Not Started | 62 |
| Not Applicable | 0 |
| Needs Investigation | 0 |
| Mitigation Implemented | 0 |
| Total | 62 |
| Total Migrated | 0 |

---

## Diagram: Diagram 1

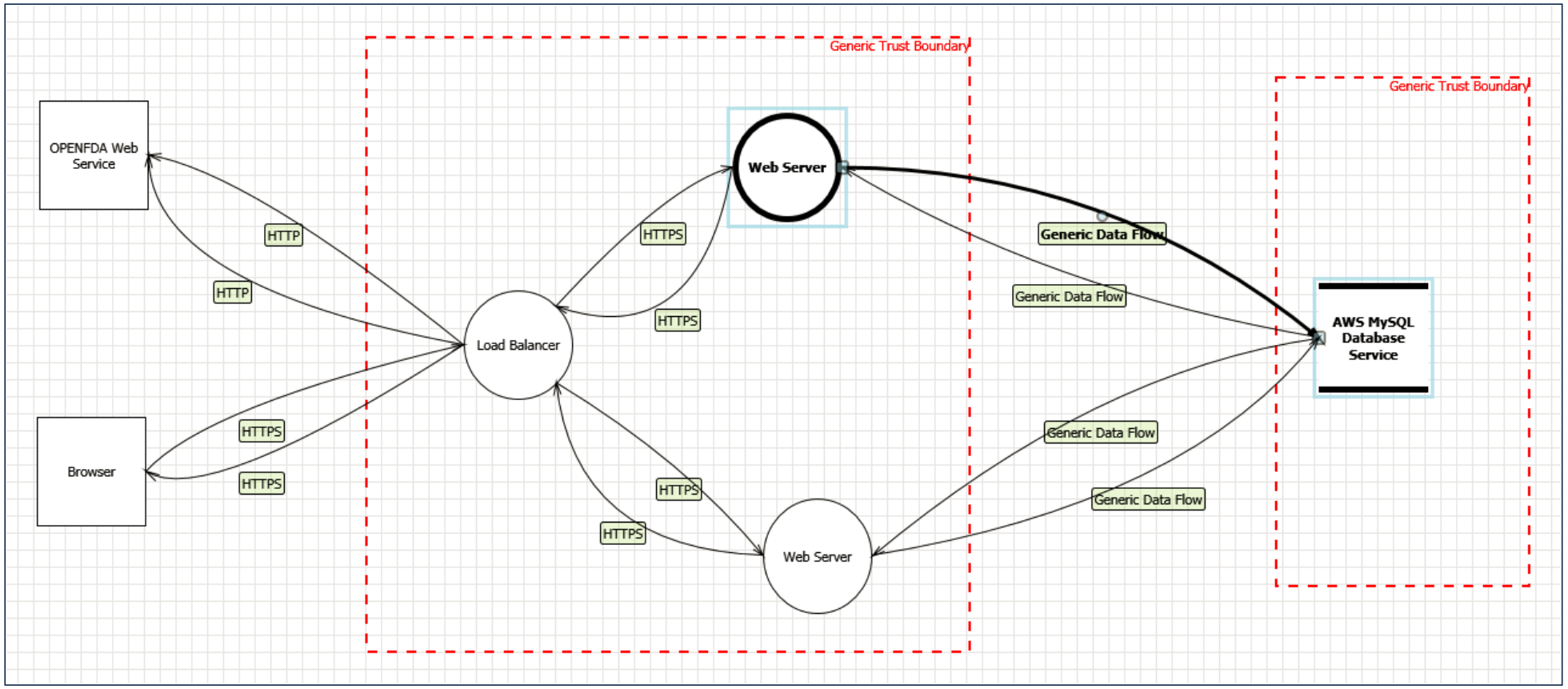
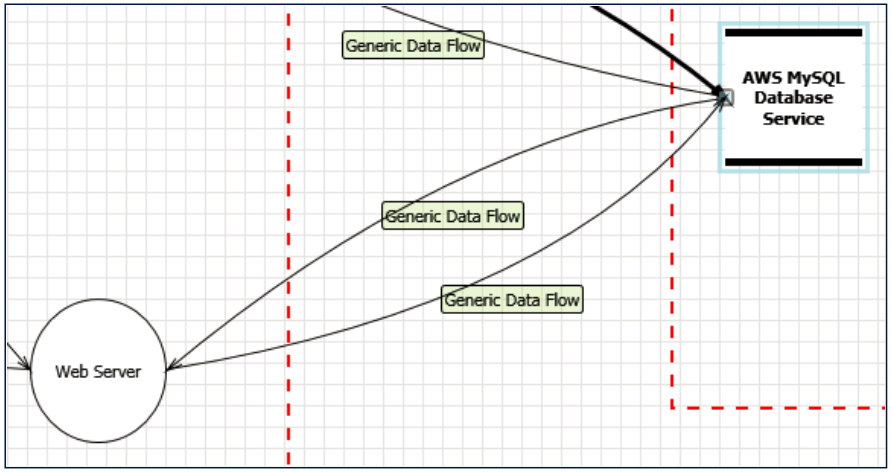



Diagram 1 Diagram Summary:

| | |
|---|---|
| Not Started | 62 |
| Not Applicable | 0 |
| Needs Investigation | 0 |
| Mitigation Implemented | 0 |
| Total | 62 |
| Total Migrated | 0 |

Interaction: Generic Data Flow




1. Elevation by Changing the Execution Flow in Web Server [State: Not Started] [Priority: High]

Category: A user subject gains increased capability or privilege by taking advantage of an implementation bug.
Description: An attacker may pass data into Web Server in order to change the flow of program execution within Web Server to the attacker's choosing.
Justification: <no mitigation provided>

2. Web Server May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Started] [Priority: High]

Category: A user subject gains increased capability or privilege by taking advantage of an implementation bug.
Description: AWS MySQL Database Service may be able to remotely execute code for Web Server.
Justification: <no mitigation provided>

3. Data Store Inaccessible [State: Not Started] [Priority: High]

Category: Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.
Description: An external agent prevents access to a data store on the other side of the trust boundary.
Justification: <no mitigation provided>

4. Data Flow Generic Data Flow Is Potentially Interrupted [State: Not Started] [Priority: High]

Category: Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.
Description: An external agent interrupts data flowing across a trust boundary in either direction.
Justification: <no mitigation provided>

5. Potential Process Crash or Stop for Web Server [State: Not Started] [Priority: High]

**Category:** Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.

**Description:** Web Server crashes, halts, stops or runs slowly; in all cases violating an availability metric.

**Justification:** <no mitigation provided>

### 6. Potential Data Repudiation by Web Server [State: Not Started] [Priority: High]

**Category:** Repudiation threats involve an adversary denying that something happened.

**Description:** Web Server claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.

**Justification:** <no mitigation provided>

### 7. Spoofing the Web Server Process [State: Not Started] [Priority: High]

**Category:** Spoofing is when a process or entity is something other than its claimed identity. Examples include substituting a process, a file, website or a network address.

**Description:** Web Server may be spoofed by an attacker and this may lead to information disclosure by AWS MySQL Database Service. Consider using a standard authentication mechanism to identify the destination process.

**Justification:** <no mitigation provided>

### 8. Spoofing of Source Data Store SQL Database [State: Not Started] [Priority: High]

**Category:** Spoofing is when a process or entity is something other than its claimed identity. Examples include substituting a process, a file, website or a network address.

**Description:** AWS MySQL Database Service may be spoofed by an attacker and this may lead to incorrect data delivered to Web Server. Consider using a standard authentication mechanism to identify the source data store.

**Justification:** <no mitigation provided>

### 9. Cross Site Scripting [State: Not Started] [Priority: High]

**Category:** Tampering is the act of altering the bits. Tampering with a process involves changing bits in the running process. Similarly, Tampering with a data flow involves changing bits on the wire or between two running processes.

**Description:** The web server 'Web Server' could be a subject to a cross-site scripting attack because it does not sanitize untrusted input.

**Justification:** <no mitigation provided>

### 10. Persistent Cross Site Scripting [State: Not Started] [Priority: High]

**Category:** Tampering is the act of altering the bits. Tampering with a process involves changing bits in the running process. Similarly, Tampering with a data flow involves changing bits on the wire or between two running processes.

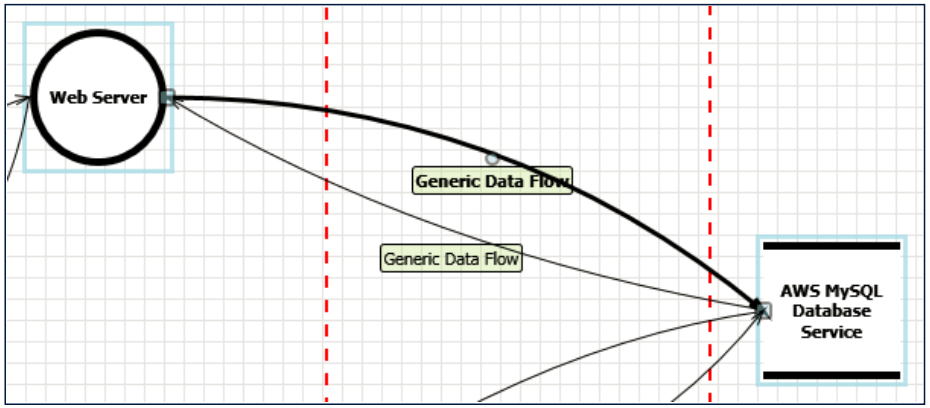
**Description:** The web server 'Web Server' could be a subject to a persistent cross-site scripting attack because it does not sanitize data store 'AWS MySQL Database Service' inputs and output.

**Justification:** <no mitigation provided>

### 11. Weak Access Control for a Resource [State: Not Started] [Priority: High]

**Category:** Information disclosure happens when the information can be read by an unauthorized party.
**Description:** Improper data protection of AWS MySQL Database Service can allow an attacker to read information not intended for disclosure. Review authorization settings.
**Justification:** <no mitigation provided>

## Interaction: Generic Data Flow




### 12. Elevation by Changing the Execution Flow in Web Server [State: Not Started] [Priority: High]

**Category:** A user subject gains increased capability or privilege by taking advantage of an implementation bug.
**Description:** An attacker may pass data into Web Server in order to change the flow of program execution within Web Server to the attacker's choosing.
**Justification:** <no mitigation provided>

### 13. Web Server May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Started] [Priority: High]

**Category:** A user subject gains increased capability or privilege by taking advantage of an implementation bug.
**Description:** AWS MySQL Database Service may be able to remotely execute code for Web Server.
**Justification:** <no mitigation provided>

### 14. Data Store Inaccessible [State: Not Started] [Priority: High]

**Category:** Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.
**Description:** An external agent prevents access to a data store on the other side of the trust boundary.
**Justification:** <no mitigation provided>

### 15. Data Flow Generic Data Flow Is Potentially Interrupted [State: Not Started] [Priority: High]

**Category:** Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.

Description: An external agent interrupts data flowing across a trust boundary in either direction.
Justification: <no mitigation provided>

16. Potential Process Crash or Stop for Web Server [State: Not Started] [Priority: High]

Category: Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.
Description: Web Server crashes, halts, stops or runs slowly; in all cases violating an availability metric.
Justification: <no mitigation provided>

17. Potential Data Repudiation by Web Server [State: Not Started] [Priority: High]

Category: Repudiation threats involve an adversary denying that something happened.
Description: Web Server claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.
Justification: <no mitigation provided>

18. Spoofing the Web Server Process [State: Not Started] [Priority: High]

Category: Spoofing is when a process or entity is something other than its claimed identity. Examples include substituting a process, a file, website or a network address.
Description: Web Server may be spoofed by an attacker and this may lead to information disclosure by AWS MySQL Database Service. Consider using a standard authentication mechanism to identify the destination process.
Justification: <no mitigation provided>

19. Spoofing of Source Data Store SQL Database [State: Not Started] [Priority: High]

Category: Spoofing is when a process or entity is something other than its claimed identity. Examples include substituting a process, a file, website or a network address.
Description: AWS MySQL Database Service may be spoofed by an attacker and this may lead to incorrect data delivered to Web Server. Consider using a standard authentication mechanism to identify the source data store.
Justification: <no mitigation provided>

20. Cross Site Scripting [State: Not Started] [Priority: High]

Category: Tampering is the act of altering the bits. Tampering with a process involves changing bits in the running process. Similarly, Tampering with a data flow involves changing bits on the wire or between two running processes.
Description: The web server 'Web Server' could be a subject to a cross-site scripting attack because it does not sanitize untrusted input.
Justification: <no mitigation provided>

21. Persistent Cross Site Scripting [State: Not Started] [Priority: High]

Category: Tampering is the act of altering the bits. Tampering with a process involves changing bits in the running process. Similarly, Tampering with a data flow involves changing bits on the wire or between two running processes.

Description: The web server 'Web Server' could be a subject to a persistent cross-site scripting attack because it does not sanitize data store 'AWS MySQL Database Service' inputs and output.

Justification: <no mitigation provided>

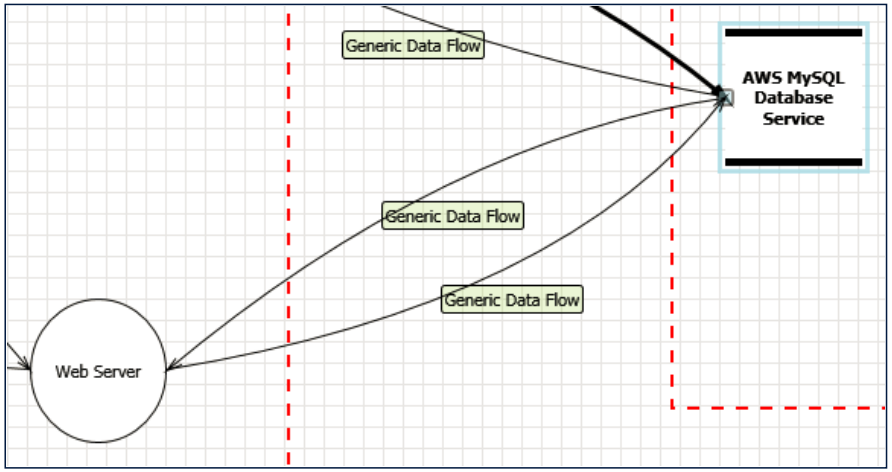
22. Weak Access Control for a Resource [State: Not Started] [Priority: High]

Category: Information disclosure happens when the information can be read by an unauthorized party.

Description: Improper data protection of AWS MySQL Database Service can allow an attacker to read information not intended for disclosure. Review authorization settings.

Justification: <no mitigation provided>

## Interaction: Generic Data Flow




23. Data Store Inaccessible [State: Not Started] [Priority: High]

Category: Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.

Description: An external agent prevents access to a data store on the other side of the trust boundary.

Justification: <no mitigation provided>

24. Data Flow Generic Data Flow Is Potentially Interrupted [State: Not Started] [Priority: High]

Category: Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.

Description: An external agent interrupts data flowing across a trust boundary in either direction.

Justification: <no mitigation provided>

25. Data Flow Sniffing [State: Not Started] [Priority: High]

**Category:** Information disclosure happens when the information can be read by an unauthorized party.

**Description:** Data flowing across Generic Data Flow may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.

**Justification:** <no mitigation provided>

### 26. Spoofing of Destination Data Store SQL Database [State: Not Started] [Priority: High]

**Category:** Spoofing is when a process or entity is something other than its claimed identity. Examples include substituting a process, a file, website or a network address.

**Description:** AWS MySQL Database Service may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of AWS MySQL Database Service. Consider using a standard authentication mechanism to identify the destination data store.

**Justification:** <no mitigation provided>

### 27. Data Store Denies SQL Database Potentially Writing Data [State: Not Started] [Priority: High]

**Category:** Repudiation threats involve an adversary denying that something happened.

**Description:** AWS MySQL Database Service claims that it did not write data received from an entity on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.

**Justification:** <no mitigation provided>

### 28. The SQL Database Data Store Could Be Corrupted [State: Not Started] [Priority: High]

**Category:** Tampering is the act of altering the bits. Tampering with a process involves changing bits in the running process. Similarly, Tampering with a data flow involves changing bits on the wire or between two running processes.

**Description:** Data flowing across Generic Data Flow may be tampered with by an attacker. This may lead to corruption of AWS MySQL Database Service. Ensure the integrity of the data flow to the data store.

**Justification:** <no mitigation provided>

### 29. Spoofing the Web Server Process [State: Not Started] [Priority: High]

**Category:** Spoofing is when a process or entity is something other than its claimed identity. Examples include substituting a process, a file, website or a network address.

**Description:** Web Server may be spoofed by an attacker and this may lead to unauthorized access to AWS MySQL Database Service. Consider using a standard authentication mechanism to identify the source process.

**Justification:** <no mitigation provided>

### 30. Potential SQL Injection Vulnerability for SQL Database [State: Not Started] [Priority: High]

**Category:** Tampering is the act of altering the bits. Tampering with a process involves changing bits in the running process. Similarly, Tampering with a data flow involves changing bits on the wire or between two running processes.

**Description:** SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives. Even parameterized data can be manipulated by a skilled and determined attacker.

Justification: <no mitigation provided>

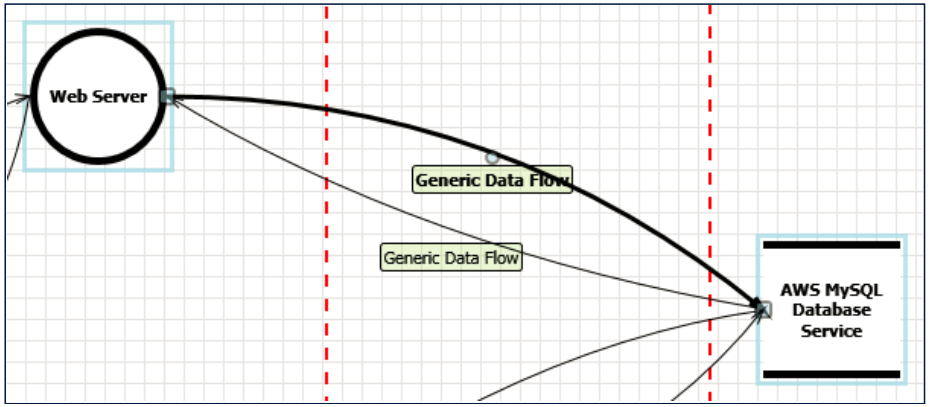
31. Potential Excessive Resource Consumption for Web Server or SQL Database [State: Not Started] [Priority: High]

Category: Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.

Description: Does Web Server or AWS MySQL Database Service take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.

Justification: <no mitigation provided>

## Interaction: Generic Data Flow




32. Data Store Inaccessible [State: Not Started] [Priority: High]

Category: Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.

Description: An external agent prevents access to a data store on the other side of the trust boundary.

Justification: <no mitigation provided>

33. Data Flow Generic Data Flow Is Potentially Interrupted [State: Not Started] [Priority: High]

Category: Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.

Description: An external agent interrupts data flowing across a trust boundary in either direction.

Justification: <no mitigation provided>

34. Data Flow Sniffing [State: Not Started] [Priority: High]

Category: Information disclosure happens when the information can be read by an unauthorized party.

Description: Data flowing across Generic Data Flow may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.

Justification: <no mitigation provided>

### 35. Data Store Denies SQL Database Potentially Writing Data [State: Not Started] [Priority: High]

Category: Repudiation threats involve an adversary denying that something happened.

Description: AWS MySQL Database Service claims that it did not write data received from an entity on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.

Justification: <no mitigation provided>

### 36. The SQL Database Data Store Could Be Corrupted [State: Not Started] [Priority: High]

Category: Tampering is the act of altering the bits. Tampering with a process involves changing bits in the running process. Similarly, Tampering with a data flow involves changing bits on the wire or between two running processes.

Description: Data flowing across Generic Data Flow may be tampered with by an attacker. This may lead to corruption of AWS MySQL Database Service. Ensure the integrity of the data flow to the data store.

Justification: <no mitigation provided>

### 37. Spoofing the Web Server Process [State: Not Started] [Priority: High]

Category: Spoofing is when a process or entity is something other than its claimed identity. Examples include substituting a process, a file, website or a network address.

Description: Web Server may be spoofed by an attacker and this may lead to unauthorized access to AWS MySQL Database Service. Consider using a standard authentication mechanism to identify the source process.

Justification: <no mitigation provided>

### 38. Spoofing of Destination Data Store SQL Database [State: Not Started] [Priority: High]

Category: Spoofing is when a process or entity is something other than its claimed identity. Examples include substituting a process, a file, website or a network address.

Description: AWS MySQL Database Service may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of AWS MySQL Database Service. Consider using a standard authentication mechanism to identify the destination data store.

Justification: <no mitigation provided>

### 39. Potential SQL Injection Vulnerability for SQL Database [State: Not Started] [Priority: High]

Category: Tampering is the act of altering the bits. Tampering with a process involves changing bits in the running process. Similarly, Tampering with a data flow involves changing bits on the wire or between two running processes.

Description: SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives. Even parameterized data can be manipulated by a skilled and determined attacker.

Justification: <no mitigation provided>

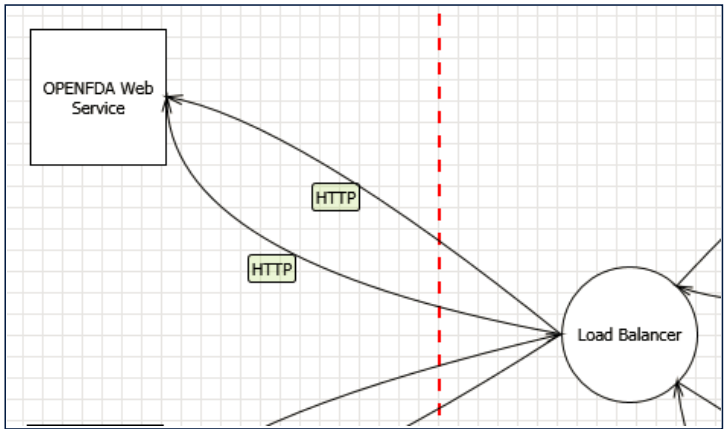
### 40. Potential Excessive Resource Consumption for Web Server or SQL Database [State: Not Started] [Priority: High]

**Category:** Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.

**Description:** Does Web Server or AWS MySQL Database Service take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.

**Justification:** <no mitigation provided>

## Interaction: HTTP




### 41. Spoofing of the OPENFDA Web Service External Destination Entity [State: Not Started] [Priority: High]

**Category:** Spoofing is when a process or entity is something other than its claimed identity. Examples include substituting a process, a file, website or a network address.

**Description:** OPENFDA Web Service may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of OPENFDA Web Service. Consider using a standard authentication mechanism to identify the external entity.

**Justification:** <no mitigation provided>

### 42. External Entity OPENFDA Web Service Potentially Denies Receiving Data [State: Not Started] [Priority: High]

**Category:** Repudiation threats involve an adversary denying that something happened.

**Description:** OPENFDA Web Service claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.

**Justification:** <no mitigation provided>

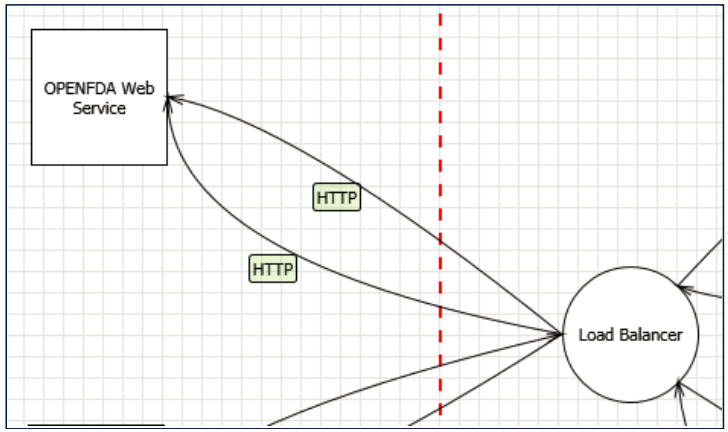
### 43. Data Flow HTTP Is Potentially Interrupted [State: Not Started] [Priority: High]

**Category:** Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.

**Description:** An external agent interrupts data flowing across a trust boundary in either direction.

**Justification:** <no mitigation provided>

## Interaction: HTTP




### 44. Spoofing of the OPENFDA Web Service External Destination Entity [State: Not Started] [Priority: High]

**Category:** Spoofing is when a process or entity is something other than its claimed identity. Examples include substituting a process, a file, website or a network address.

**Description:** OPENFDA Web Service may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of OPENFDA Web Service. Consider using a standard authentication mechanism to identify the external entity.

**Justification:** <no mitigation provided>

### 45. External Entity OPENFDA Web Service Potentially Denies Receiving Data [State: Not Started] [Priority: High]

**Category:** Repudiation threats involve an adversary denying that something happened.

**Description:** OPENFDA Web Service claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.

**Justification:** <no mitigation provided>

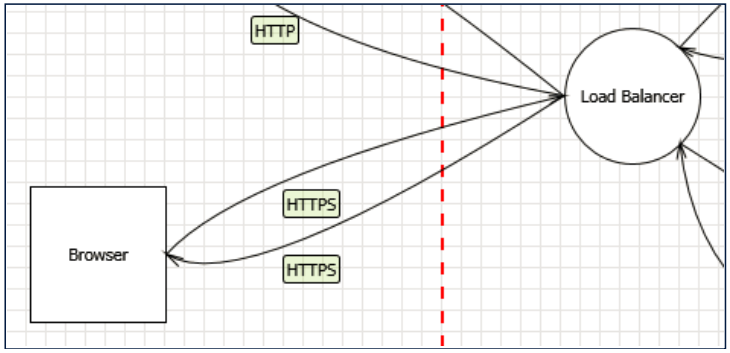
### 46. Data Flow HTTP Is Potentially Interrupted [State: Not Started] [Priority: High]

**Category:** Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.

**Description:** An external agent interrupts data flowing across a trust boundary in either direction.

**Justification:** <no mitigation provided>

## Interaction: HTTPS




47. Elevation by Changing the Execution Flow in Generic Process [State: Not Started] [Priority: High]

**Category:** A user subject gains increased capability or privilege by taking advantage of an implementation bug.
**Description:** An attacker may pass data into Load Balancer in order to change the flow of program execution within Load Balancer to the attacker's choosing.
**Justification:** <no mitigation provided>

48. Generic Process May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Started] [Priority: High]

**Category:** A user subject gains increased capability or privilege by taking advantage of an implementation bug.
**Description:** Browser may be able to remotely execute code for Load Balancer.
**Justification:** <no mitigation provided>

49. Elevation Using Impersonation [State: Not Started] [Priority: High]

**Category:** A user subject gains increased capability or privilege by taking advantage of an implementation bug.
**Description:** Load Balancer may be able to impersonate the context of Browser in order to gain additional privilege.
**Justification:** <no mitigation provided>

50. Data Flow HTTPS Is Potentially Interrupted [State: Not Started] [Priority: High]

**Category:** Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.
**Description:** An external agent interrupts data flowing across a trust boundary in either direction.
**Justification:** <no mitigation provided>

51. Potential Process Crash or Stop for Generic Process [State: Not Started] [Priority: High]

**Category:** Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.
**Description:** Load Balancer crashes, halts, stops or runs slowly; in all cases violating an availability metric.
**Justification:** <no mitigation provided>

52. Potential Data Repudiation by Generic Process [State: Not Started] [Priority: High]

**Category:** Repudiation threats involve an adversary denying that something happened.

**Description:** Load Balancer claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.

**Justification:** <no mitigation provided>

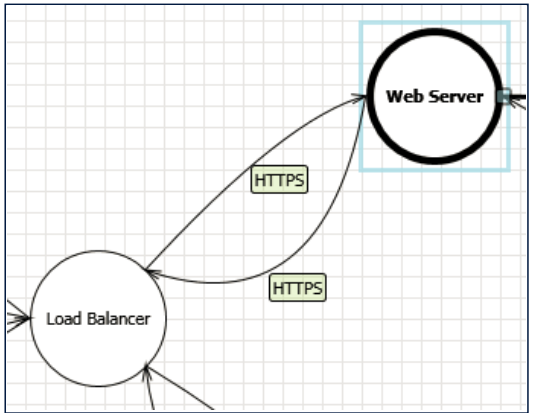
53. Spoofing the Browser External Entity [State: Not Started] [Priority: High]

**Category:** Spoofing is when a process or entity is something other than its claimed identity. Examples include substituting a process, a file, website or a network address.

**Description:** Browser may be spoofed by an attacker and this may lead to unauthorized access to Load Balancer. Consider using a standard authentication mechanism to identify the external entity.

**Justification:** <no mitigation provided>

## Interaction: HTTPS



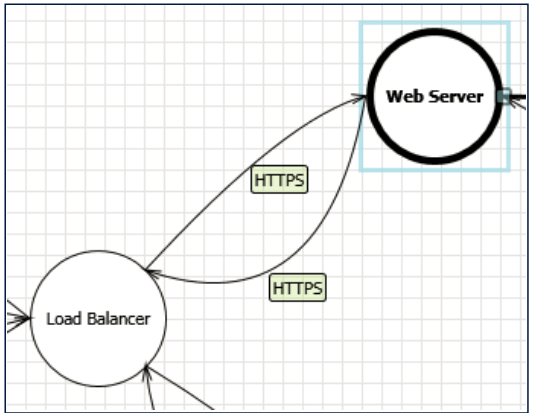

54. Elevation Using Impersonation [State: Not Started] [Priority: High]

**Category:** A user subject gains increased capability or privilege by taking advantage of an implementation bug.

**Description:** Load Balancer may be able to impersonate the context of Web Server in order to gain additional privilege.

**Justification:** <no mitigation provided>

## Interaction: HTTPS



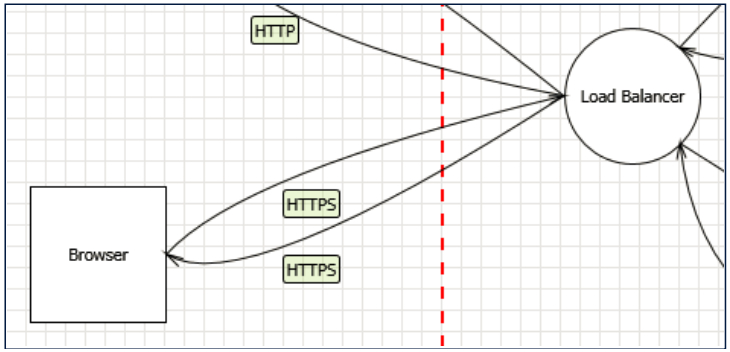

55. Elevation Using Impersonation [State: Not Started] [Priority: High]

**Category:** A user subject gains increased capability or privilege by taking advantage of an implementation bug.
**Description:** Web Server may be able to impersonate the context of Load Balancer in order to gain additional privilege.
**Justification:** <no mitigation provided>

56. Cross Site Scripting [State: Not Started] [Priority: High]

**Category:** Tampering is the act of altering the bits. Tampering with a process involves changing bits in the running process. Similarly, Tampering with a data flow involves changing bits on the wire or between two running processes.
**Description:** The web server 'Web Server' could be a subject to a cross-site scripting attack because it does not sanitize untrusted input.
**Justification:** <no mitigation provided>

## Interaction: HTTPS




57. Spoofing of the Browser External Destination Entity [State: Not Started] [Priority: High]

Category: Spoofing is when a process or entity is something other than its claimed identity. Examples include substituting a process, a file, website or a network address.

Description: Browser may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of Browser. Consider using a standard authentication mechanism to identify the external entity.

Justification: <no mitigation provided>

58. External Entity Browser Potentially Denies Receiving Data [State: Not Started] [Priority: High]

Category: Repudiation threats involve an adversary denying that something happened.

Description: Browser claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.

Justification: <no mitigation provided>

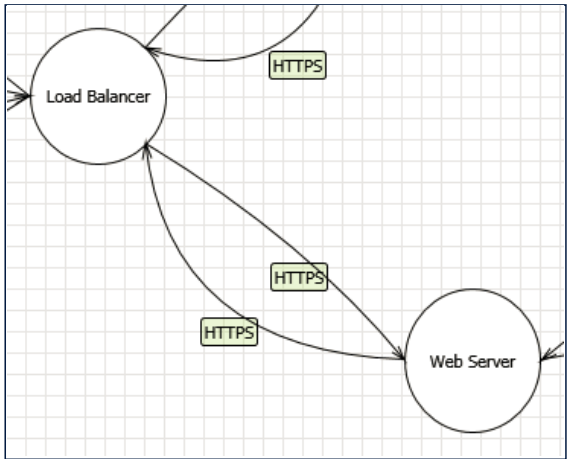
59. Data Flow HTTPS Is Potentially Interrupted [State: Not Started] [Priority: High]

Category: Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.

Description: An external agent interrupts data flowing across a trust boundary in either direction.

Justification: <no mitigation provided>

## Interaction: HTTPS




60. Cross Site Scripting [State: Not Started] [Priority: High]

Category: Tampering is the act of altering the bits. Tampering with a process involves changing bits in the running process. Similarly, Tampering with a data flow involves changing bits on the wire or between two running processes.

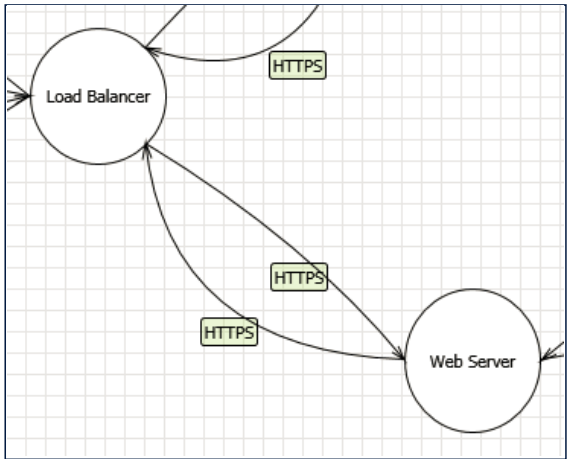
Description: The web server 'Web Server' could be a subject to a cross-site scripting attack because it does not sanitize untrusted input.

Justification: <no mitigation provided>

61. Elevation Using Impersonation [State: Not Started] [Priority: High]

**Category:** A user subject gains increased capability or privilege by taking advantage of an implementation bug.
**Description:** Web Server may be able to impersonate the context of Load Balancer in order to gain additional privilege.
**Justification:** <no mitigation provided>

## Interaction: HTTPS




62. Elevation Using Impersonation [State: Not Started] [Priority: High]

**Category:** A user subject gains increased capability or privilege by taking advantage of an implementation bug.
**Description:** Load Balancer may be able to impersonate the context of Web Server in order to gain additional privilege.
**Justification:** <no mitigation provided>