Threat Modeling Report

Created on 07-05-2023 21:11:32

Threat Model Name:

Owner:

Reviewer:

Contributors:

Description:

Assumptions:

External Dependencies:

Threat Model Summary:

Not Started	77
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	77
Total Migrated	0

Diagram: Diagram 1

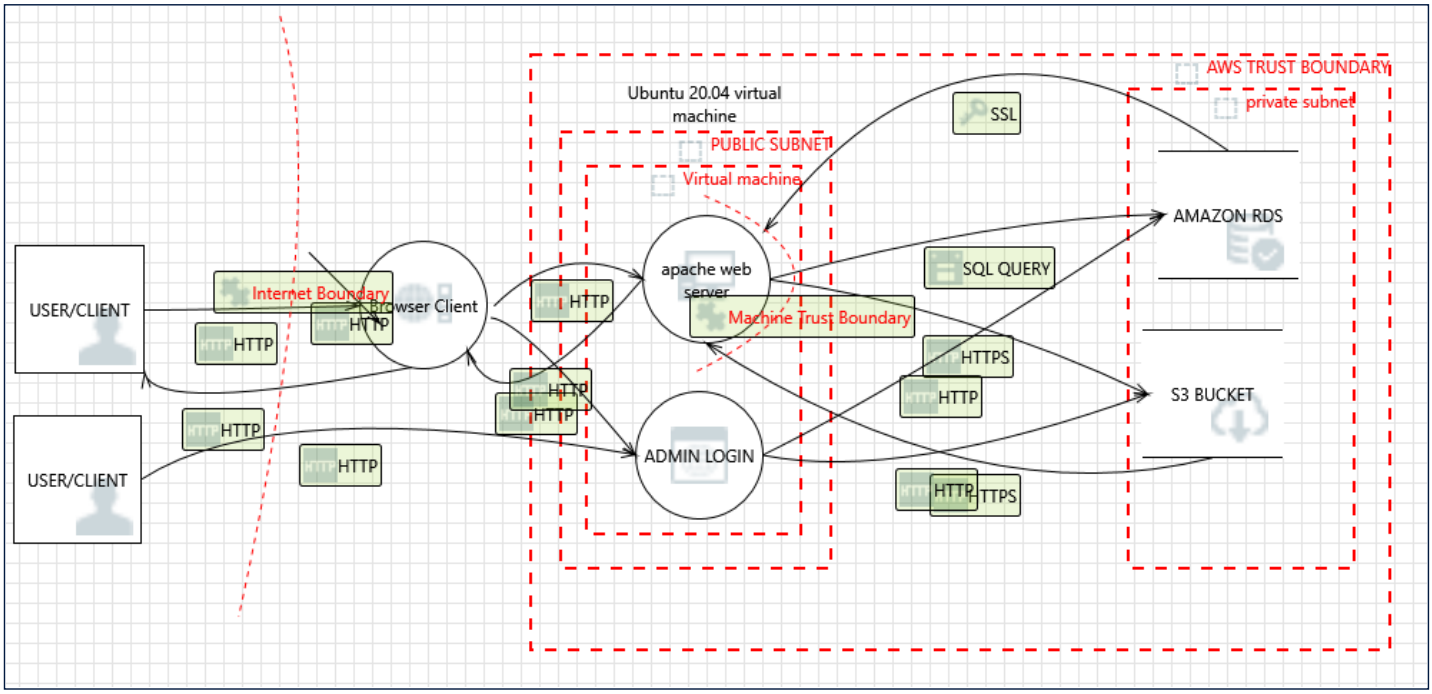
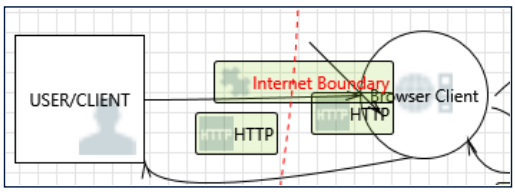


Diagram 1 Diagram Summary:

Not Started	77
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	77
Total Migrated	0

Interaction: HTTP



1. Spoofing the USER/CLIENT External Entity [State: Not Started] [Priority: High]

Category: Spoofing
Description: USER/CLIENT may be spoofed by an attacker and this may lead to unauthorized access to Browser Client. Consider using a standard authentication mechanism to identify the external entity.
Justification: <no mitigation provided>

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

Category: Elevation Of Privilege
Description: Browser Client may be able to impersonate the context of USER/CLIENT in order to gain additional privilege.
Justification: <no mitigation provided>

3. Spoofing the Browser Client Process [State: Not Started] [Priority: High]

Category: Spoofing
Description: Browser Client may be spoofed by an attacker and this may lead to information disclosure by USER/CLIENT. Consider using a standard authentication mechanism to identify the destination process.
Justification: <no mitigation provided>

4. Potential Lack of Input Validation for Browser Client [State: Not Started] [Priority: High]

Category: Tampering
Description: Data flowing across HTTP may be tampered with by an attacker. This may lead to a denial of service attack against Browser Client or an elevation of privilege attack against Browser Client or an information disclosure by Browser Client. Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.
Justification: <no mitigation provided>

5. Potential Data Repudiation by Browser Client [State: Not Started] [Priority: High]

Category: Repudiation
Description: Browser Client 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>

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

Category: Information Disclosure
Description: Data flowing across HTTP 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>

7. Potential Process Crash or Stop for Browser Client [State: Not Started] [Priority: High]

Category: Denial Of Service
Description: Browser Client crashes, halts, stops or runs slowly; in all cases violating an availability metric.
Justification: <no mitigation provided>

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

Category: Denial Of Service

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

Justification: <no mitigation provided>

9. Browser Client May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Started] [Priority: High]

Category: Elevation Of Privilege

Description: USER/CLIENT may be able to remotely execute code for Browser Client.

Justification: <no mitigation provided>

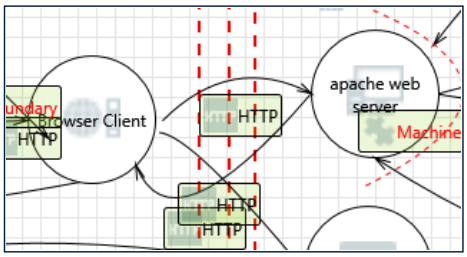
10. Elevation by Changing the Execution Flow in Browser Client [State: Not Started] [Priority: High]

Category: Elevation Of Privilege

Description: An attacker may pass data into Browser Client in order to change the flow of program execution within Browser Client to the attacker's choosing.

Justification: <no mitigation provided>

Interaction: HTTP



11. Spoofing the apache web server Process [State: Not Started] [Priority: High]

Category: Spoofing

Description: apache web server may be spoofed by an attacker and this may lead to unauthorized access to Browser Client. Consider using a standard authentication mechanism to identify the source process.

Justification: <no mitigation provided>

12. Spoofing the Browser Client Process [State: Not Started] [Priority: High]

Category: Spoofing

Description: Browser Client may be spoofed by an attacker and this may lead to information disclosure by apache web server. Consider using a standard authentication mechanism to identify the destination process.

Justification: <no mitigation provided>

13. Potential Lack of Input Validation for Browser Client [State: Not Started] [Priority: High]

Category: Tampering

Description: Data flowing across HTTP may be tampered with by an attacker. This may lead to a denial of service attack against Browser Client or an elevation of privilege attack against Browser Client or an information disclosure by Browser Client. Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.

Justification: <no mitigation provided>

14. apache web server Process Memory Tampered [State: Not Started] [Priority: High]

Category: Tampering

Description: If apache web server is given access to memory, such as shared memory or pointers, or is given the ability to control what Browser Client executes (for example, passing back a function pointer.), then apache web server can tamper with Browser Client. Consider if the function could work with less access to memory, such as passing data rather than pointers. Copy in data provided, and then validate it.

Justification: <no mitigation provided>

15. Potential Data Repudiation by Browser Client [State: Not Started] [Priority: High]

Category: Repudiation

Description: Browser Client 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>

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

Category: Information Disclosure

Description: Data flowing across HTTP 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>

17. Potential Process Crash or Stop for Browser Client [State: Not Started] [Priority: High]

Category: Denial Of Service

Description: Browser Client crashes, halts, stops or runs slowly; in all cases violating an availability metric.

Justification: <no mitigation provided>

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

Category: Denial Of Service

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

Justification: <no mitigation provided>

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

Category: Elevation Of Privilege

Description: Browser Client may be able to impersonate the context of apache web server in order to gain additional privilege.

Justification: <no mitigation provided>

20. Browser Client May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Started] [Priority: High]

Category: Elevation Of Privilege

Description: apache web server may be able to remotely execute code for Browser Client.

Justification: <no mitigation provided>

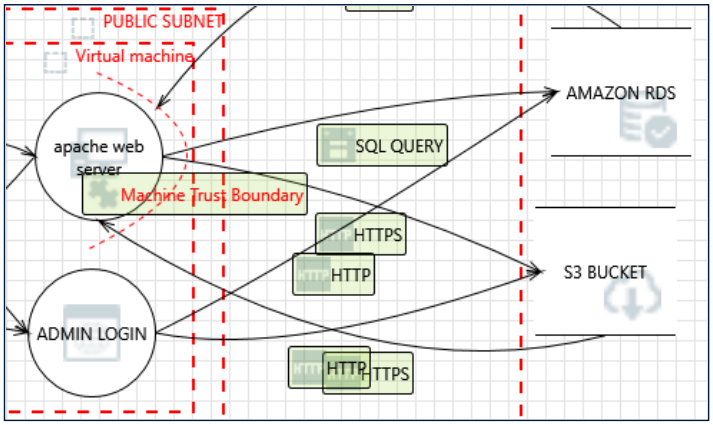
21. Elevation by Changing the Execution Flow in Browser Client [State: Not Started] [Priority: High]

Category: Elevation Of Privilege

Description: An attacker may pass data into Browser Client in order to change the flow of program execution within Browser Client to the attacker's choosing.

Justification: <no mitigation provided>

Interaction: HTTP



22. Spoofing of Destination Data Store AMAZON RDS [State: Not Started] [Priority: High]

Category: Spoofing
Description: AMAZON RDS may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of AMAZON RDS. Consider using a standard authentication mechanism to identify the destination data store.
Justification: <no mitigation provided>

23. Potential SQL Injection Vulnerability for AMAZON RDS [State: Not Started] [Priority: High]

Category: Tampering
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>

24. The AMAZON RDS Data Store Could Be Corrupted [State: Not Started] [Priority: High]

Category: Tampering
Description: Data flowing across HTTP may be tampered with by an attacker. This may lead to corruption of AMAZON RDS. Ensure the integrity of the data flow to the data store.
Justification: <no mitigation provided>

25. Data Store Denies AMAZON RDS Potentially Writing Data [State: Not Started] [Priority: High]

Category: Repudiation
Description: AMAZON RDS 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>

26. Authorization Bypass [State: Not Started] [Priority: High]

Category: Information Disclosure
Description: Can you access AMAZON RDS and bypass the permissions for the object? For example by editing the files directly with a hex editor, or reaching it via filesharing? Ensure that your program is the only one that can access the data, and that all other subjects have to use your interface.
Justification: <no mitigation provided>

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

Category: Information Disclosure
Description: Data flowing across HTTP 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>

28. Potential Excessive Resource Consumption for ADMIN LOGIN or AMAZON RDS [State: Not Started] [Priority: High]

Category: Denial Of Service
Description: Does ADMIN LOGIN or AMAZON RDS 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>

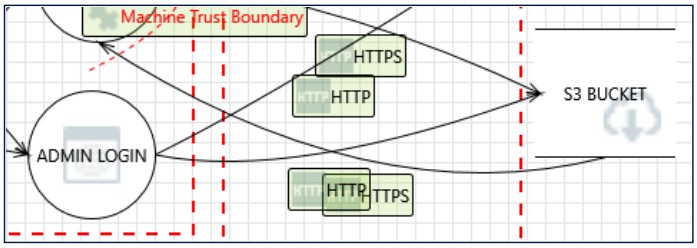
29. Data Flow HTTP Is Potentially Interrupted [State: Not Started] [Priority: High]

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

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

Category: Denial Of Service
Description: An external agent prevents access to a data store on the other side of the trust boundary.
Justification: <no mitigation provided>

Interaction: HTTP



31. Spoofing of Destination Data Store S3 BUCKET [State: Not Started] [Priority: High]

Category: Spoofing
Description: S3 BUCKET may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of S3 BUCKET. Consider using a standard authentication mechanism to identify the destination data store.
Justification: <no mitigation provided>

32. The S3 BUCKET Data Store Could Be Corrupted [State: Not Started] [Priority: High]

Category: Tampering
Description: Data flowing across HTTP may be tampered with by an attacker. This may lead to corruption of S3 BUCKET. Ensure the integrity of the data flow to the data store.
Justification: <no mitigation provided>

33. Data Store Denies S3 BUCKET Potentially Writing Data [State: Not Started] [Priority: High]

Category: Repudiation
Description: S3 BUCKET 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>

34. Authorization Bypass [State: Not Started] [Priority: High]

Category: Information Disclosure
Description: Can you access S3 BUCKET and bypass the permissions for the object? For example by editing the files directly with a hex editor, or reaching it via filesharing? Ensure that your program is the only one that can access the data, and that all other subjects have to use your interface.
Justification: <no mitigation provided>

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

Category: Information Disclosure
Description: Data flowing across HTTP 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>

36. Potential Excessive Resource Consumption for ADMIN LOGIN or S3 BUCKET [State: Not Started] [Priority: High]

Category: Denial Of Service
Description: Does ADMIN LOGIN or S3 BUCKET 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>

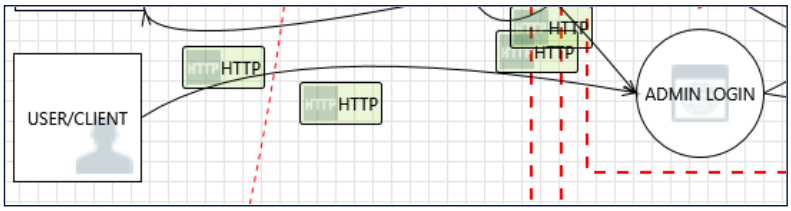
37. Data Flow HTTP Is Potentially Interrupted [State: Not Started] [Priority: High]

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

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

Category: Denial Of Service
Description: An external agent prevents access to a data store on the other side of the trust boundary.
Justification: <no mitigation provided>

Interaction: HTTP



39. Spoofing the ADMIN LOGIN Process [State: Not Started] [Priority: High]

Category: Spoofing
Description: ADMIN LOGIN may be spoofed by an attacker and this may lead to information disclosure by USER/CLIENT. Consider using a standard authentication mechanism to identify the destination process.
Justification: <no mitigation provided>

40. Potential Lack of Input Validation for ADMIN LOGIN [State: Not Started] [Priority: High]

Category: Tampering
Description: Data flowing across HTTP may be tampered with by an attacker. This may lead to a denial of service attack against ADMIN LOGIN or an elevation of privilege attack against ADMIN LOGIN or an information disclosure by ADMIN LOGIN. Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.
Justification: <no mitigation provided>

41. Potential Data Repudiation by ADMIN LOGIN [State: Not Started] [Priority: High]

Category: Repudiation
Description: ADMIN LOGIN 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>

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

Category: Information Disclosure
Description: Data flowing across HTTP 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>

43. Potential Process Crash or Stop for ADMIN LOGIN [State: Not Started] [Priority: High]

Category: Denial Of Service
Description: ADMIN LOGIN crashes, halts, stops or runs slowly; in all cases violating an availability metric.
Justification: <no mitigation provided>

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

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

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

Category: Elevation Of Privilege

Description: ADMIN LOGIN may be able to impersonate the context of USER/CLIENT in order to gain additional privilege.
Justification: <no mitigation provided>

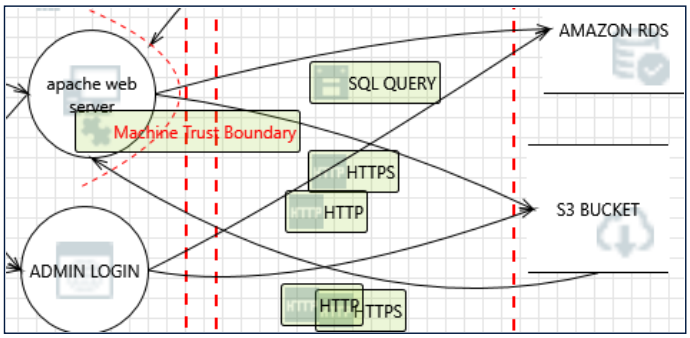
46. ADMIN LOGIN May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Started] [Priority: High]

Category: Elevation Of Privilege
Description: USER/CLIENT may be able to remotely execute code for ADMIN LOGIN.
Justification: <no mitigation provided>

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

Category: Elevation Of Privilege
Description: An attacker may pass data into ADMIN LOGIN in order to change the flow of program execution within ADMIN LOGIN to the attacker's choosing.
Justification: <no mitigation provided>

Interaction: HTTPS



48. Elevation by Changing the Execution Flow in apache web server [State: Not Started] [Priority: High]

Category: Elevation Of Privilege
Description: An attacker may pass data into apache web server in order to change the flow of program execution within apache web server to the attacker's choosing.
Justification: <no mitigation provided>

49. apache web server May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Started] [Priority: High]

Category: Elevation Of Privilege
Description: S3 BUCKET may be able to remotely execute code for apache web server.
Justification: <no mitigation provided>

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

Category: Denial Of Service
Description: An external agent prevents access to a data store on the other side of the trust boundary.
Justification: <no mitigation provided>

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

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

52. Potential Process Crash or Stop for apache web server [State: Not Started] [Priority: High]

Category: Denial Of Service
Description: apache web server crashes, halts, stops or runs slowly; in all cases violating an availability metric.
Justification: <no mitigation provided>

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

Category: Information Disclosure
Description: Improper data protection of S3 BUCKET can allow an attacker to read information not intended for disclosure. Review authorization settings.
Justification: <no mitigation provided>

54. Potential Data Repudiation by apache web server [State: Not Started] [Priority: High]

Category: Repudiation
Description: apache 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>

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

Category: Tampering
Description: The web server 'apache web server' could be a subject to a persistent cross-site scripting attack because it does not sanitize data store 'S3 BUCKET' inputs and output.
Justification: <no mitigation provided>

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

Category: Tampering
Description: The web server 'apache web server' could be a subject to a cross-site scripting attack because it does not sanitize untrusted input.
Justification: <no mitigation provided>

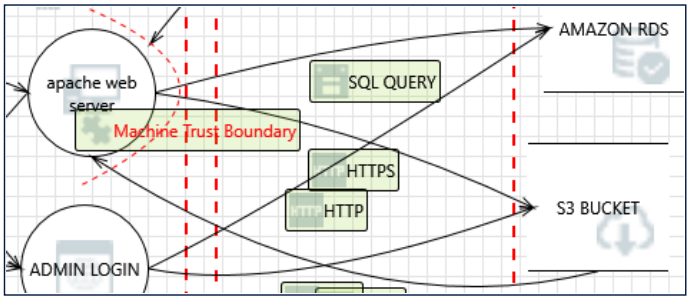
57. Spoofing of Source Data Store S3 BUCKET [State: Not Started] [Priority: High]

Category: Spoofing
Description: S3 BUCKET may be spoofed by an attacker and this may lead to incorrect data delivered to apache web server. Consider using a standard authentication mechanism to identify the source data store.
Justification: <no mitigation provided>

58. Spoofing the apache web server Process [State: Not Started] [Priority: High]

Category: Spoofing
Description: apache web server may be spoofed by an attacker and this may lead to information disclosure by S3 BUCKET. Consider using a standard authentication mechanism to identify the destination process.
Justification: <no mitigation provided>

Interaction: HTTPS



59. Spoofing the apache web server Process [State: Not Started] [Priority: High]

Category: Spoofing
Description: apache web server may be spoofed by an attacker and this may lead to unauthorized access to S3 BUCKET. Consider using a standard authentication mechanism to identify the source process.
Justification: <no mitigation provided>

60. Spoofing of Destination Data Store S3 BUCKET [State: Not Started] [Priority: High]

Category: Spoofing
Description: S3 BUCKET may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of S3 BUCKET. Consider using a standard authentication mechanism to identify the destination data store.
Justification: <no mitigation provided>

61. The S3 BUCKET Data Store Could Be Corrupted [State: Not Started] [Priority: High]

Category: Tampering
Description: Data flowing across HTTPS may be tampered with by an attacker. This may lead to corruption of S3 BUCKET. Ensure the integrity of the data flow to the data store.
Justification: <no mitigation provided>

62. Data Store Denies S3 BUCKET Potentially Writing Data [State: Not Started] [Priority: High]

Category: Repudiation
Description: S3 BUCKET 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>

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

Category: Information Disclosure
Description: Data flowing across HTTPS 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>

64. Potential Excessive Resource Consumption for apache web server or S3 BUCKET [State: Not Started] [Priority: High]

Category: Denial Of Service
Description: Does apache web server or S3 BUCKET 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>

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

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

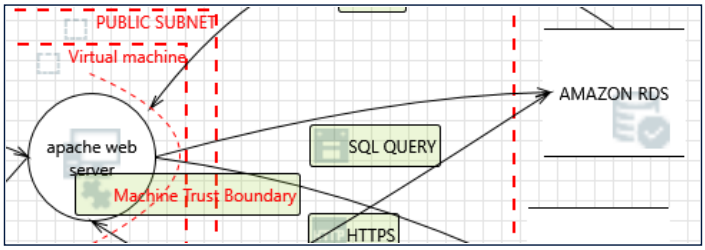
66. Data Store Inaccessible [State: Not Started] [Priority: High]

Category: Denial Of Service
Description: An external agent prevents access to a data store on the other side of the trust boundary.
Justification: <no mitigation provided>

67. Weak Credential Transit [State: Not Started] [Priority: High]

Category: Information Disclosure
Description: Credentials on the wire are often subject to sniffing by an attacker. Are the credentials re-usable/re-playable? Are credentials included in a message? For example, sending a zip file with the password in the email. Use strong cryptography for the transmission of credentials. Use the OS libraries if at all possible, and consider cryptographic algorithm agility, rather than hardcoding a choice.
Justification: <no mitigation provided>

Interaction: SQL QUERY



68. Spoofing the apache web server Process [State: Not Started] [Priority: High]

Category: Spoofing
Description: apache web server may be spoofed by an attacker and this may lead to unauthorized access to AMAZON RDS. Consider using a standard authentication mechanism to identify the source process.
Justification: <no mitigation provided>

69. Spoofing of Destination Data Store AMAZON RDS [State: Not Started] [Priority: High]

Category: Spoofing
Description: AMAZON RDS may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of AMAZON RDS. Consider using a standard authentication mechanism to identify the destination data store.
Justification: <no mitigation provided>

70. Authenticated Data Flow Compromised [State: Not Started] [Priority: High]

Category: Tampering
Description: An attacker can read or modify data transmitted over an authenticated dataflow.
Justification: <no mitigation provided>

71. Potential SQL Injection Vulnerability for AMAZON RDS [State: Not Started] [Priority: High]

Category: Tampering
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>

72. The AMAZON RDS Data Store Could Be Corrupted [State: Not Started] [Priority: High]

Category: Tampering
Description: Data flowing across SQL QUERY may be tampered with by an attacker. This may lead to corruption of AMAZON RDS. Ensure the integrity of the data flow to the data store.
Justification: <no mitigation provided>

73. Data Store Denies AMAZON RDS Potentially Writing Data [State: Not Started] [Priority: High]

Category: Repudiation
Description: AMAZON RDS 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>

74. Potential Excessive Resource Consumption for apache web server or AMAZON RDS [State: Not Started] [Priority: High]

Category: Denial Of Service
Description: Does apache web server or AMAZON RDS 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>

75. Data Flow SQL QUERY Is Potentially Interrupted [State: Not Started] [Priority: High]

Category: Denial Of Service

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

Justification: <no mitigation provided>

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

Category: Denial Of Service

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

Justification: <no mitigation provided>

77. Weak Credential Transit [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: Credentials on the wire are often subject to sniffing by an attacker. Are the credentials re-usable/re-playable? Are credentials included in a message? For example, sending a zip file with the password in the email. Use strong cryptography for the transmission of credentials. Use the OS libraries if at all possible, and consider cryptographic algorithm agility, rather than hardcoding a choice.

Justification: <no mitigation provided>