Threat Modeling Report

Created on 11/19/2024 12:01:54 AM

Threat Model Name:

Owner:

Reviewer:

Contributors:

Description:

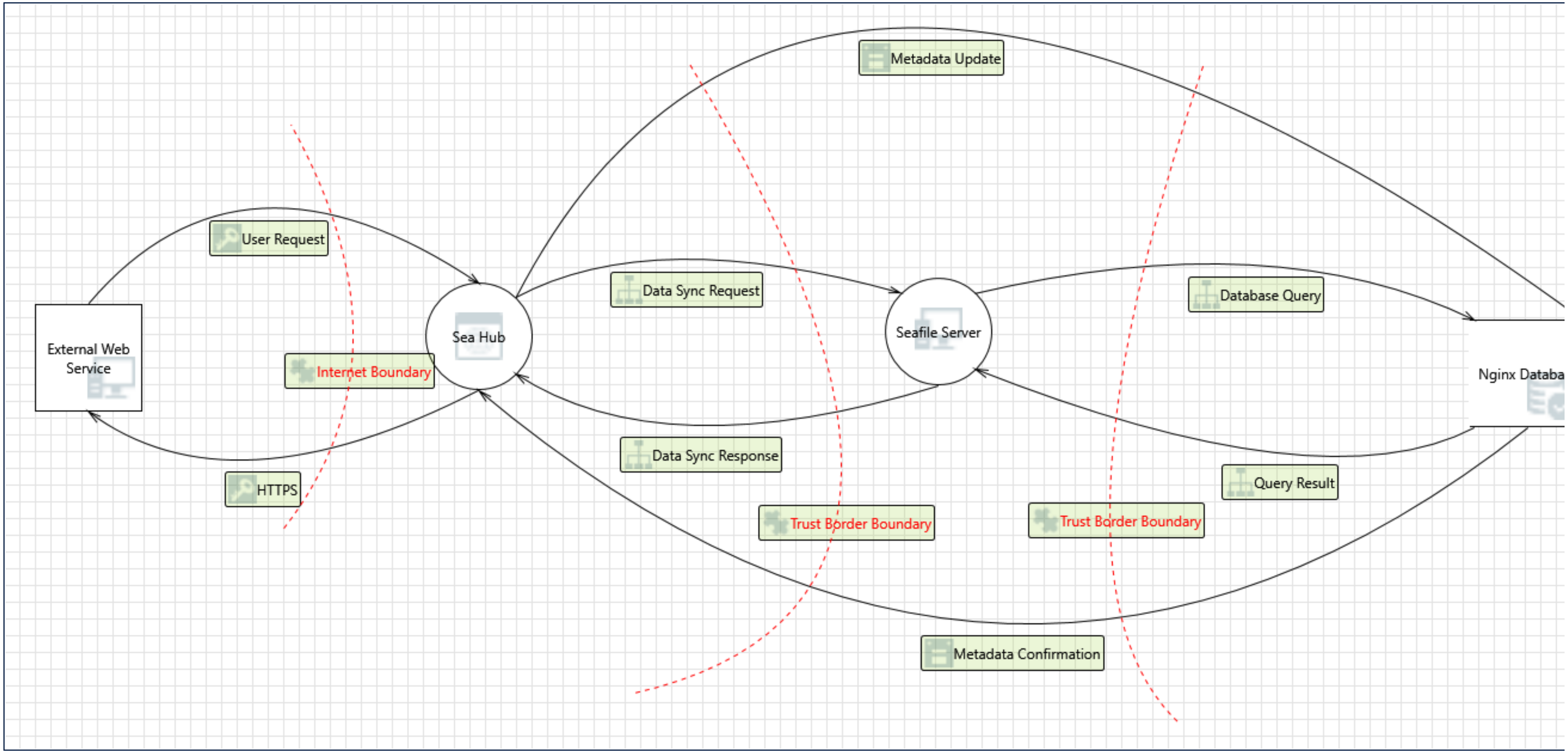
Assumptions:

External Dependencies:

Threat Model Summary:

Not Started	6
Not Applicable	9
Needs Investigation	39
Mitigation Implemented	21
Total	75
Total Migrated	0

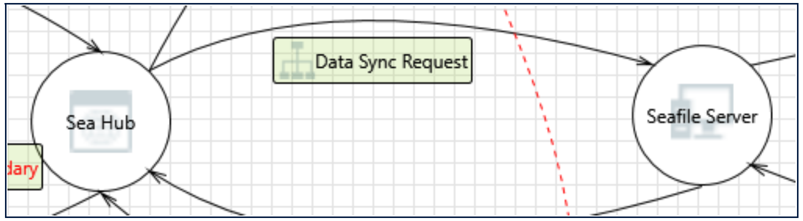
Diagram: Seafile DFD



Seafile DFD Diagram Summary:

Not Started	6
Not Applicable	9
Needs Investigation	39
Mitigation Implemented	21
Total	75
Total Migrated	0

Interaction: Data Sync Request



1. Elevation Using Impersonation [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: Seafile Server may be able to impersonate the context of Sea Hub in order to gain additional privilege.

Justification: The use of roles limits access, checking these permissions as different actons are called further limit attempts

2. Cross Site Scripting [State: Mitigation Implemented] [Priority: High]

Category: Tampering

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

Justification: HTML will be escaped, sanitization of javascript and validations of input such as email and zipcode will be used

3. Cross Site Request Forgery [State: Needs Investigation] [Priority: High]

Category: Elevation Of Privilege

Description: Cross-site request forgery (CSRF or XSRF) is a type of attack in which an attacker forces a user's browser to make a forged request to a vulnerable site by exploiting an existing trust relationship between the browser and the vulnerable web site. In a simple scenario, a user is logged in to web site A using a cookie as a credential. The other browses to web site B. Web site B returns a page with a hidden form that posts to web site A. Since the browser will carry the user's cookie to web site A, web site B now can take any action on web site A, for example, adding an admin to an account. The attack can be used to exploit any requests that the browser automatically authenticates, e.g. by session cookie, integrated authentication, IP whitelisting. The attack can be carried out in many ways such as by luring the victim to a site under control of the attacker, getting the user to click a link in a phishing email, or hacking a reputable web site that the victim will visit. The issue can only be resolved on the server side by requiring that all authenticated state-changing requests include an additional piece of secret payload (canary or CSRF token) which is known only to the legitimate web site and the browser and which is protected in transit through SSL/TLS. See the Forgery Protection property on the flow stencil for a list of mitigations.

Justification: Need to properly utilize tokens and cookies. HTML will be escaped, sanitization of javascript and validations of input such as email and zipcode will be used

4. Elevation by Changing the Execution Flow in Seafile Server [State: Needs Investigation] [Priority: High]

Category: Elevation Of Privilege

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

Justification: The use of roles limits access, checking these permissions as different actons are called further limit attempts

5. Seafile Server May be Subject to Elevation of Privilege Using Remote Code Execution [State: Needs Investigation] [Priority: High]

Category: Elevation Of Privilege

Description: Sea Hub may be able to remotely execute code for Seafile Server.

Justification: The use of roles limits access, checking these permissions as different actons are called further limit attempts

6. Data Flow Data Sync Request Is Potentially Interrupted [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

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

Justification: Stress testing can be used to determine the robustness of the application. May show lack of memory or processing power, log capacity etc....

7. Potential Process Crash or Stop for Seafile Server [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

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

Justification: Stress testing can be used to determine the robustness of the application. May show lack of memory or processing power, log capacity etc....

8. Data Flow Sniffing [State: Mitigation Implemented] [Priority: High]

Category: Information Disclosure

Description: Data flowing across Data Sync Request 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: Utilization of secure protocols (HTTPS, SFTP and SSH) will prevent sniffing.

9. Potential Data Repudiation by Seafile Server [State: Not Started] [Priority: High]

Category: Repudiation

Description: Seafile 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: Need to ensure logging and auditing are adequate to provide the needed level of details

10. Potential Lack of Input Validation for Seafile Server [State: Mitigation Implemented] [Priority: High]

Category: Tampering

Description: Data flowing across Data Sync Request may be tampered with by an attacker. This may lead to a denial of service attack against Seafile Server or an elevation of privilege attack against Seafile Server or an information disclosure by Seafile Server. 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: HTML will be escaped, sanitization of javascript and validations of input such as email and zipcode will be used

11. Spoofing the Seafile Server Process [State: Not Applicable] [Priority: High]

Category: Spoofing

Description: Seafile Server may be spoofed by an attacker and this may lead to information disclosure by Sea Hub. Consider using a standard authentication mechanism to identify the destination process.

Justification: With the trust boundaries, this should not occur

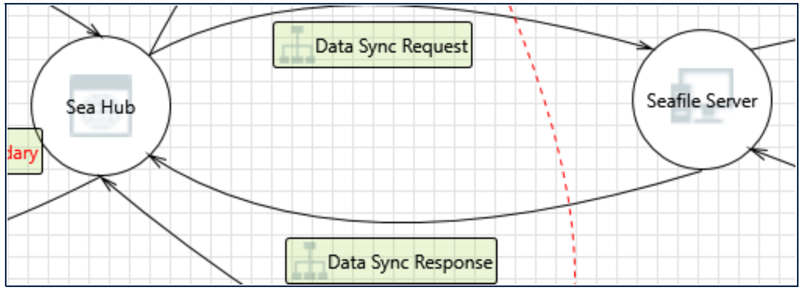
12. Spoofing the Sea Hub Process [State: Not Applicable] [Priority: High]

Category: Spoofing

Description: Sea Hub may be spoofed by an attacker and this may lead to unauthorized access to Seafile Server. Consider using a standard authentication mechanism to identify the source process.

Justification: With the trust boundaries, this should not occur

Interaction: Data Sync Response



13. Seafile Server Process Memory Tampered [State: Needs Investigation] [Priority: Low]

Category: Tampering

Description: If Seafile Server is given access to memory, such as shared memory or pointers, or is given the ability to control what Sea Hub executes (for example, passing back a function pointer.), then Seafile Server can tamper with Sea Hub. 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: security boundary is provided

14. Cross Site Scripting [State: Needs Investigation] [Priority: Medium]

Category: Tampering

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

Justification: SeaHub&#39;s reliance on Django offers inherent XSS protections, improper coding practices, lack of sanitization, or unsafe template rendering could introduce vulnerabilities.

15. Elevation Using Impersonation [State: Not Applicable] [Priority: Medium]

Category: Elevation Of Privilege

Description: Sea Hub may be able to impersonate the context of Seafile Server in order to gain additional privilege.

Justification: Seafile&#39;s architecture, when configured properly, minimizes this risk through role-based controls, tokenized authentication, and encrypted communication. Regular security audits and updates are crucial to ensure no vulnerabilities allow such an escalation

16. Elevation by Changing the Execution Flow in Sea Hub [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

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

Justification: The use of roles limits access, checking these permissions as different actons are called further limit attempts

17. Sea Hub May be Subject to Elevation of Privilege Using Remote Code Execution [State: Needs Investigation] [Priority: High]

Category: Elevation Of Privilege

Description: Seafile Server may be able to remotely execute code for Sea Hub.

Justification: The use of roles limits access, checking these permissions as different actons are called further limit attempts

18. Data Flow Data Sync Response Is Potentially Interrupted [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

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

Justification: Stress testing can be used to determine the robustness of the application. May show lack of memory or processing power, log capacity etc....

19. Potential Process Crash or Stop for Sea Hub [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

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

Justification: Stress testing can be used to determine the robustness of the application. May show lack of memory or processing power, log capacity etc....

20. Data Flow Sniffing [State: Needs Investigation] [Priority: High]

Category: Information Disclosure

Description: Data flowing across Data Sync Response 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: Need to ensure logging and auditing are adequate to provide the needed level of details

21. Potential Data Repudiation by Sea Hub [State: Needs Investigation] [Priority: High]

Category: Repudiation

Description: Sea Hub 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: Need to ensure logging and auditing are adequate to provide the needed level of details

22. Potential Lack of Input Validation for Sea Hub [State: Mitigation Implemented] [Priority: High]

Category: Tampering

Description: Data flowing across Data Sync Response may be tampered with by an attacker. This may lead to a denial of service attack against Sea Hub or an elevation of privilege attack against Sea Hub or an information disclosure by Sea Hub. 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: Utilization of secure protocols (HTTPS, SFTP and SSH) will prevent sniffing.

23. Spoofing the Sea Hub Process [State: Not Started] [Priority: High]

Category: Spoofing

Description: Sea Hub may be spoofed by an attacker and this may lead to information disclosure by Seafile Server. Consider using a standard authentication mechanism to identify the destination process.

Justification: User authentication combined with roles will limit this risk. Limiting network access via ports and IP address can further ensure protections

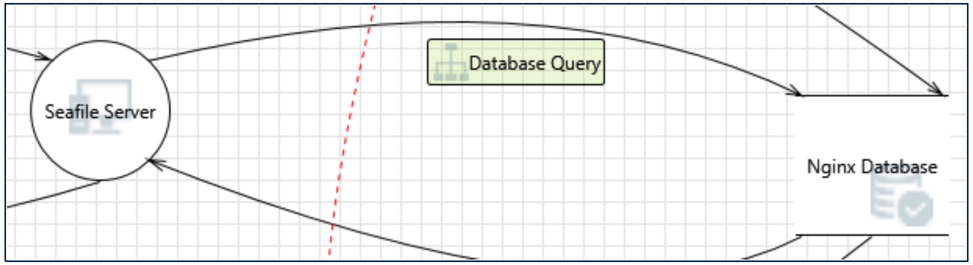
24. Spoofing the Seafile Server Process [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Seafile Server may be spoofed by an attacker and this may lead to unauthorized access to Sea Hub. Consider using a standard authentication mechanism to identify the source process.

Justification: User authentication combined with roles will limit this risk. Limiting network access via ports and IP address can further ensure protections

Interaction: Database Query



25. Spoofing of Destination Data Store Nginx Database [State: Needs Investigation] [Priority: High]

Category: Spoofing

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

Justification: Further investigation is required to ascertain the claim

26. Potential SQL Injection Vulnerability for Nginx Database [State: Not Applicable] [Priority: Low]

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: Seafile&#39;s use of modern development practices like ORMs and parameterized queries addresses most SQL injection risks.

27. Potential Excessive Resource Consumption for Seafile Server or Nginx Database [State: Mitigation Implemented] [Priority: Low]

Category: Denial Of Service

Description: Does Seafile Server or Nginx Database 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: Seafile Server: Explicitly controls storage usage and sync operations through configuration options.
Nginx: Explicitly manages connections, rate limits, bandwidth, and caching to ensure optimal server performance.

28. Spoofing the Seafile Server Process [State: Needs Investigation] [Priority: High]

Category: Spoofing

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

Justification: User authentication combined with roles will limit this risk. Limiting network access via ports and IP address can further ensure protections

29. The Nginx Database Data Store Could Be Corrupted [State: Mitigation Implemented] [Priority: High]

Category: Tampering

Description: Data flowing across Database Query may be tampered with by an attacker. This may lead to corruption of Nginx Database. Ensure the integrity of the data flow to the data store.

Justification: With the trust boundaries, this should not occur

30. Data Store Denies Nginx Database Potentially Writing Data [State: Needs Investigation] [Priority: High]

Category: Repudiation

Description: Nginx Database 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: Need to ensure logging and auditing are adequate to provide the needed level of details

31. Data Flow Sniffing [State: Needs Investigation] [Priority: High]

Category: Information Disclosure

Description: Data flowing across Database Query 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: The use of roles limits access, checking these permissions as different actons are called further limit attempts

32. Data Flow Database Query Is Potentially Interrupted [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

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

Justification: Stress testing can be used to determine the robustness of the application. May show lack of memory or processing power, log capacity etc....

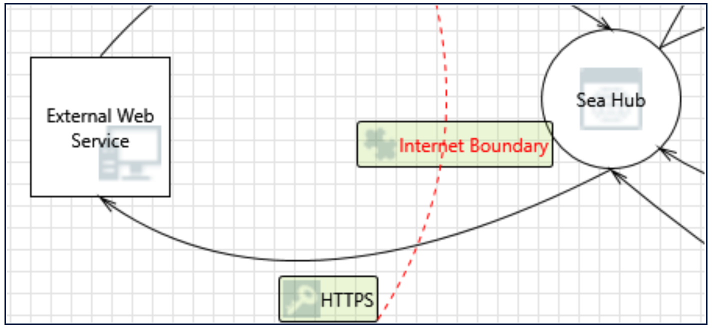
33. Data Store Inaccessible [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

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

Justification: Stress testing can be used to determine the robustness of the application. May show lack of memory or processing power, log capacity etc....

Interaction: HTTPS



34. Spoofing of the External Web Service External Destination Entity [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

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

Justification: The utilization of certificates for HTTPS allows for secure communication from the client to server.

35. External Entity External Web Service Potentially Denies Receiving Data [State: Needs Investigation] [Priority: High]

Category: Repudiation

Description: External 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: Need to ensure logging and auditing are adequate to provide the needed level of details

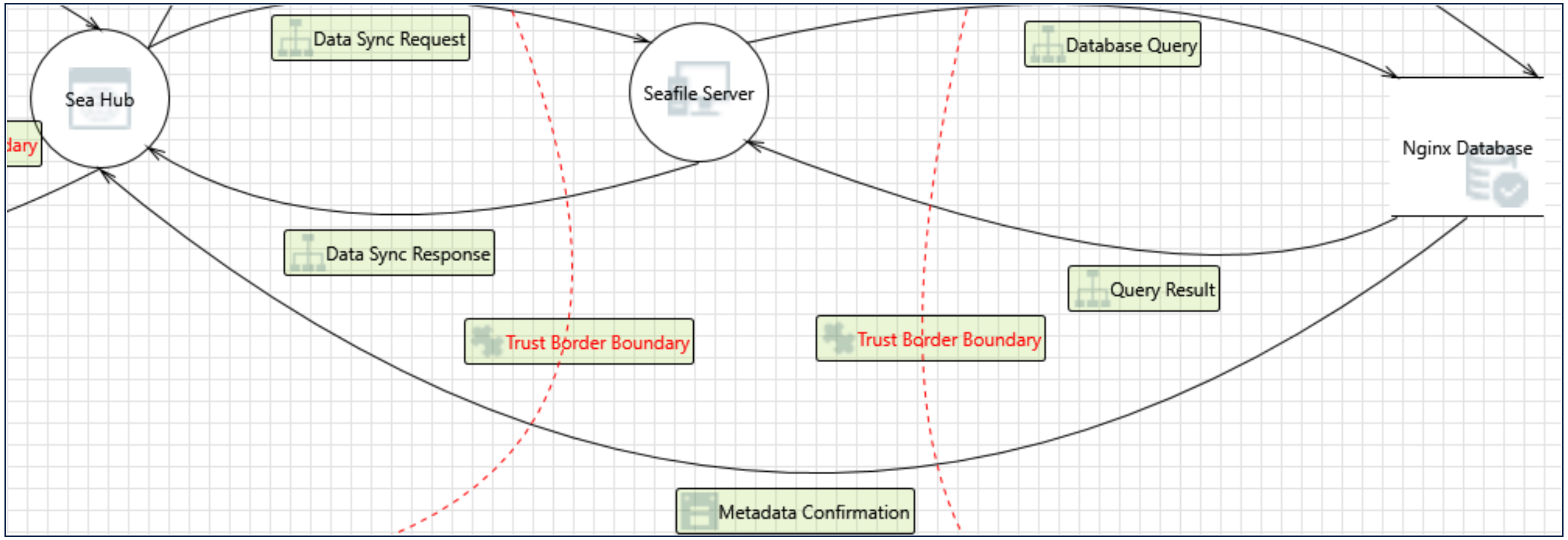
36. Data Flow HTTPS Is Potentially Interrupted [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

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

Justification: Will need to validate the ports are limited, can handle bad calls and is robust enough to handle bad calls. SSH and HTTPS will aid in preventing MiTM

Interaction: Metadata Confirmation



37. Spoofing the Sea Hub Process [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Sea Hub may be spoofed by an attacker and this may lead to information disclosure by Nginx Database. Consider using a standard authentication mechanism to identify the destination process.

Justification: Utilization of secure protocols (HTTPS, SFTP and SSH) will prevent sniffing. Plus trust boundaries

38. Spoofing of Source Data Store Nginx Database [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Nginx Database may be spoofed by an attacker and this may lead to incorrect data delivered to Sea Hub. Consider using a standard authentication mechanism to identify the source data store.

Justification: Utilization of secure protocols (HTTPS, SFTP and SSH) will prevent sniffing. Plus trust boundaries

39. Cross Site Scripting [State: Mitigation Implemented] [Priority: High]

Category: Tampering

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

Justification: Utilization of secure protocols (HTTPS, SFTP and SSH) will prevent sniffing.

40. Persistent Cross Site Scripting [State: Mitigation Implemented] [Priority: High]

Category: Tampering

Description: The web server 'Sea Hub' could be a subject to a persistent cross-site scripting attack because it does not sanitize data store 'Nginx Database' inputs and output.

Justification: Utilization of secure protocols (HTTPS, SFTP and SSH) will prevent sniffing.

41. Potential Data Repudiation by Sea Hub [State: Needs Investigation] [Priority: High]

Category: Repudiation

Description: Sea Hub 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: Need to ensure logging and auditing are adequate to provide the needed level of details

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

Category: Information Disclosure

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

Justification: The use of roles limits access, checking these permissions as different actons are called further limit attempts

43. Potential Process Crash or Stop for Sea Hub [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

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

Justification: Stress testing can be used to determine the robustness of the application. May show lack of memory or processing power, log capacity etc....

44. Data Flow Metadata Confirmation Is Potentially Interrupted [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

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

Justification: Stress testing can be used to determine the robustness of the application. May show lack of memory or processing power, log capacity etc....

45. Data Store Inaccessible [State: Needs Investigation] [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: Stress testing can be used to determine the robustness of the application. May show lack of memory or processing power, log capacity etc....

46. Sea Hub May be Subject to Elevation of Privilege Using Remote Code Execution [State: Needs Investigation] [Priority: High]

Category: Elevation Of Privilege

Description: Nginx Database may be able to remotely execute code for Sea Hub.

Justification: The use of roles limits access, checking these permissions as different actons are called further limit attempts

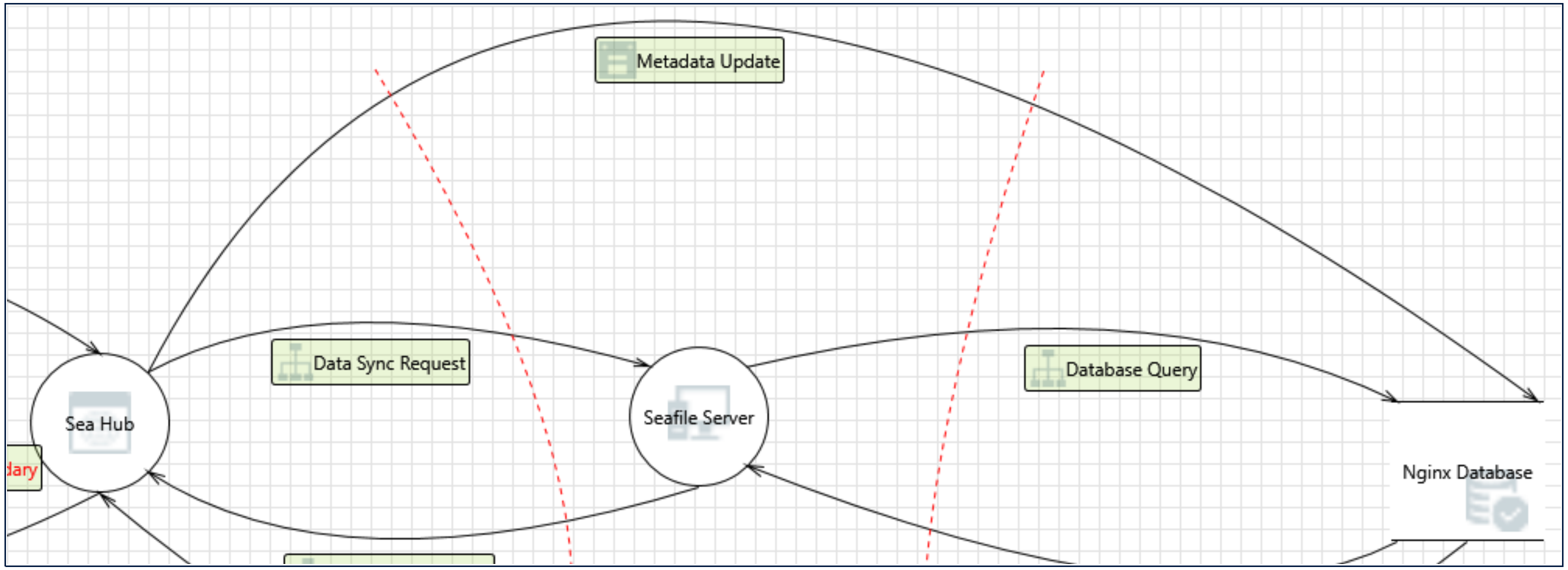
47. Elevation by Changing the Execution Flow in Sea Hub [State: Needs Investigation] [Priority: High]

Category: Elevation Of Privilege

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

Justification: The use of roles limits access, checking these permissions as different actons are called further limit attempts

Interaction: Metadata Update



48. Spoofing of Destination Data Store Nginx Database [State: Not Applicable] [Priority: High]

Category: Spoofing

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

Justification: Attacker cannot directly access this system

49. Potential SQL Injection Vulnerability for Nginx Database [State: Mitigation Implemented] [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: Work on the front end web calls will prevent this;HTML will be escaped, sanitization of javascript and validations of input such as email and zipcode will be used

50. Potential Excessive Resource Consumption for Sea Hub or Nginx Database [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

Description: Does Sea Hub or Nginx Database 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: Stress testing can be used to determine the robustness of the application. May show lack of memory or processing power, log capacity etc....

51. Spoofing the Sea Hub Process [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Sea Hub may be spoofed by an attacker and this may lead to unauthorized access to Nginx Database. Consider using a standard authentication mechanism to identify the source process.

Justification: User authentication combined with roles will limit this risk. Limiting network access via ports and IP address can further ensure protections

52. The Nginx Database Data Store Could Be Corrupted [State: Not Applicable] [Priority: High]

Category: Tampering

Description: Data flowing across Metadata Update may be tampered with by an attacker. This may lead to corruption of Nginx Database. Ensure the integrity of the data flow to the data store.

Justification: With the trust boundaries, this should not occur

53. Data Store Denies Nginx Database Potentially Writing Data [State: Needs Investigation] [Priority: High]

Category: Repudiation

Description: Nginx Database 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: Need to ensure logging and auditing are adequate to provide the needed level of details

54. Data Flow Sniffing [State: Mitigation Implemented] [Priority: High]

Category: Information Disclosure

Description: Data flowing across Metadata Update 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: Utilization of secure protocols (HTTPS, SFTP and SSH) will prevent sniffing.

55. Data Flow Metadata Update 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: Stress testing can be used to determine the robustness of the application. May show lack of memory or processing power, log capacity etc....

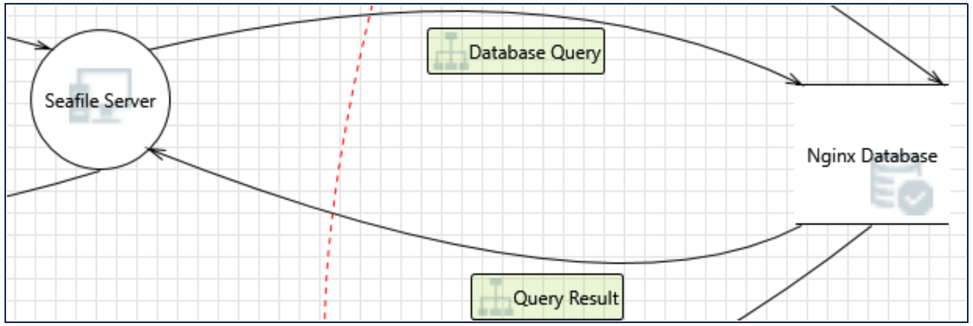
56. Data Store Inaccessible [State: Not Started] [Priority: High]

Category: Denial Of Service

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

Justification: Stress testing can be used to determine the robustness of the application. May show lack of memory or processing power, log capacity etc....

Interaction: Query Result



57. Weak Access Control for a Resource [State: Mitigation Implemented] [Priority: High]

Category: Information Disclosure

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

Justification: Utilization of roles and monitoring of logging.

58. Persistent Cross Site Scripting [State: Needs Investigation] [Priority: High]

Category: Tampering

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

Justification: further investigation required

59. Cross Site Scripting [State: Not Applicable] [Priority: Low]

Category: Tampering

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

Justification: Security boundary is in place

60. Spoofing of Source Data Store Nginx Database [State: Not Applicable] [Priority: Low]

Category: Spoofing

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

Justification: The risk of an attacker spoofing a database and delivering incorrect data to Seafile Server is mitigated with proper configurations, including encrypted database connections, strong authentication, and a secure Nginx setup

61. Spoofing the Seafile Server Process [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Seafile Server may be spoofed by an attacker and this may lead to information disclosure by Nginx Database. Consider using a standard authentication mechanism to identify the destination process.

Justification: Utilization of secure protocols (HTTPS, SFTP and SSH) will prevent sniffing. Plus trust boundaries

62. Potential Data Repudiation by Seafile Server [State: Needs Investigation] [Priority: High]

Category: Repudiation

Description: Seafile 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: Need to ensure logging and auditing are adequate to provide the needed level of details

63. Potential Process Crash or Stop for Seafile Server [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

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

Justification: Stress testing can be used to determine the robustness of the application. May show lack of memory or processing power, log capacity etc....

64. Data Flow Query Result Is Potentially Interrupted [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

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

Justification: Stress testing can be used to determine the robustness of the application. May show lack of memory or processing power, log capacity etc....

65. Data Store Inaccessible [State: Needs Investigation] [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: Stress testing can be used to determine the robustness of the application. May show lack of memory or processing power, log capacity etc....

66. Seafile Server May be Subject to Elevation of Privilege Using Remote Code Execution [State: Needs Investigation] [Priority: High]

Category: Elevation Of Privilege

Description: Nginx Database may be able to remotely execute code for Seafile Server.

Justification: The use of roles limits access, checking these permissions as different actons are called further limit attempts

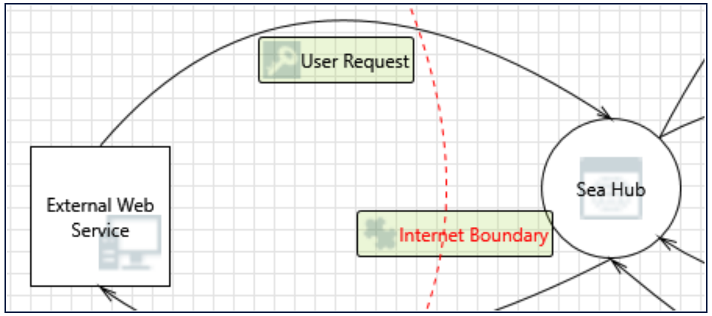
67. Elevation by Changing the Execution Flow in Seafile Server [State: Needs Investigation] [Priority: High]

Category: Elevation Of Privilege

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

Justification: The use of roles limits access, checking these permissions as different actons are called further limit attempts

Interaction: User Request



68. Spoofing the External Web Service External Entity [State: Mitigation Implemented] [Priority: Medium]

Category: Spoofing

Description: External Web Service may be spoofed by an attacker and this may lead to unauthorized access to Sea Hub. Consider using a standard authentication mechanism to identify the external entity.

Justification: Users must login with MFA credential

69. Cross Site Scripting [State: Mitigation Implemented] [Priority: Low]

Category: Tampering

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

Justification: User inputs are sanitized

70. Elevation Using Impersonation [State: Not Applicable] [Priority: Low]

Category: Elevation Of Privilege

Description: Sea Hub may be able to impersonate the context of External Web Service in order to gain additional privilege.

Justification: Seahub is not a threat to external web service

71. Elevation by Changing the Execution Flow in Sea Hub [State: Needs Investigation] [Priority: High]

Category: Elevation Of Privilege

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

Justification: The use of roles limits access, checking these permissions as different actons are called further limit attempts

72. Sea Hub May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Started] [Priority: High]

Category: Elevation Of Privilege

Description: External Web Service may be able to remotely execute code for Sea Hub.

Justification: The use of roles limits access, checking these permissions as different actons are called further limit attempts

73. Data Flow User Request Is Potentially Interrupted [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

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

Justification: Stress testing can be used to determine the robustness of the application. May show lack of memory or processing power, log capacity etc....

74. Potential Process Crash or Stop for Sea Hub [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

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

Justification: Vulnerability assesment and stress testing can be used to determine the robustness of the applicatin

75. Potential Data Repudiation by Sea Hub [State: Needs Investigation] [Priority: High]

Category: Repudiation

Description: Sea Hub 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: Need to ensure logging and auditing are adequate to provide the needed level of details