Aircraft Service Scenario

Assignment 3
SENG 360

Amaan Makhani
Tashinga Zvawada
WeiJun Syu

For the threat report

Not Applicable" means that the threat was deemed not a problem while "Mitigated" means that it was deemed a problem and a potential solution was proposed.

1, 22- Human interactors.- The system requires human interactors namely the office support and maintenance crew. Both of these elements will represent all the human factors that might be part of the system being designed.

5: Crew Certifications- Datastore in the form of an SQL database. This is because the data allows management data stored in the database. Most of the data regarding the crew certifications will be related to a number of records that need clearance hence applying an SQL database works well since it is a relational database and will have all the information related to the other in the databases.

6,7 : Service records and Checklist and Manuals- This is a datastore in the form of an SQL database to store the data related to Service records and Checklist and manuals. The database is SQL since the data found in the database is related to each other in terms of the different records contained in the datastore. A relational database would ensure easy retrieval of related information and easy updating of the data found in both of the databases.

14: Service applications - represented by the process in the form of a native application since this application is intended to work on Apple iPads which run on iOS. The native application element offers the best representation of our threat model. It fits the description of the required application which was that the service application is intended to run on Apple iPads.Being native also allows it to use device specific hardware like cache.

20,21: Cache datastore to store some of the service records, checklist and manuals on the iPad since some of the data needs to be internal to the ipad and cache memory offers an efficient way to store and communicate data to the service application.

8,9,10,11,12,13: HTTPS dataflow-This is to ensure secure communication between the database and the service application over the local network. to represent secure flow of requests and responses between the service application and the databases

16,17,18,19: Dataflow between the cache holding the service record and checklist and manuals and the service application. This represents the basic cache to application interaction inside the Apple iPads.

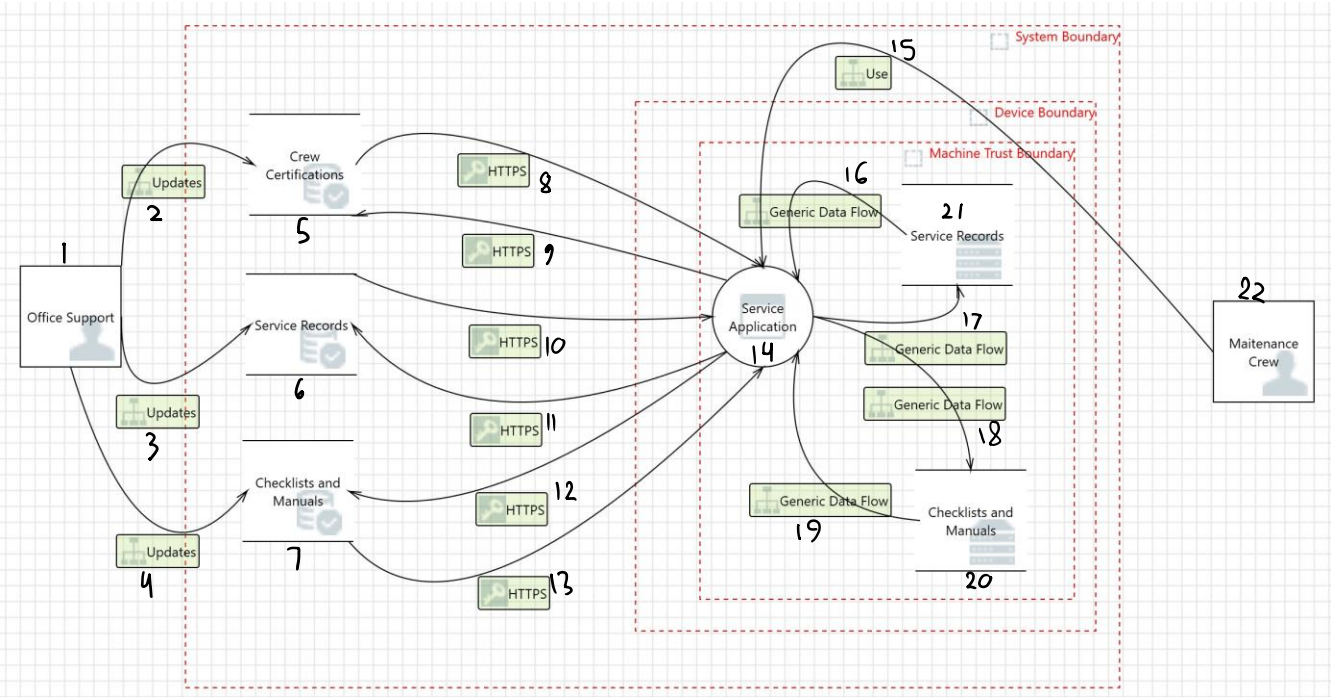
2,3,4: Update dataflow: to represent how the office support would communicate with the databases inside the System boundary. The office support would usually interact with the database by updating the data stored in them. We assume that there is a response from the database of interaction.

15: Use dataflow- For the maintenance crew to use the application the way they intend to use it. We assume that the application responds in the form of the GUI of the ipad.

Machine trust and device trust boundary represents the device used by the human interactor that is the iPad. This is why we have the service application together with the cache datastore inside this boundary to clearly show that the interaction is on the same device and anything outside these will not be part of the iPad.

The System Boundary- represents the environment in which all the elements will be interacting in order for the processes to run and the data to flow between these elements.or the service application to run. It is the best way to represent the organization's wifi in which the service application will be used to access the databases.

Property Explanations



For each element, look at and adjust its properties, as necessary. For each property, give a 1-2 sentence description of why you set that property that way.

1: For the human interactor we set the authenticates itself to yes. We did this as we know for the office staff to have access to the databases, they would first have to login through their company computers therefore already authenticating themselves.

2, 3, and 4: For the data flow from the office support to the databases we assume this will be using Wi-Fi as many companies avoid wired connections now as it limits flexibility. We also set the source to be pre authenticated as information is coming from the office computers, however, we left the destination authentication as is since we do not believe that sending the packets will guarantee we make it to the databases. We could have our information intercepted so therefore we chose not to select an authenticated destination.

5: Since crew certifications are important, we decided to deem these credentials as they prove the crew's qualifications. We assumed this data will be encrypted based on the presence of a data center this is a fact that would not be overlooked. Since crew certifications will be modified, added, or deleted we set the database to write access. Due to the importance of the data being stored and the liability it prevents we assume the company has a backup of the database.

6 and 7: Since the service records and manuals contain an instructional/logged information we set the store log data property to yes. To prevent other customers or unauthorized users to read the data we assumed this data will be encrypted as this is of most importance. Since both service records and manuals are added, and removed we set the write access property to yes as this will be needed for the database to serve its function. Due to the importance of the data being stored and the liability it prevents we assume the company has a backup of the databases.

8, 10, and 13: For our https data process we assumed it will be using Wi-Fi as that only made sense since the iPads are portable no other network is possible. It is also stated they are on the local network. When coming from the databases we assumed the source is authenticated as the databases will be originating the information from a static data center. These databases will also have unique identifiers so these should allow for authentication of the source.

9, 11, and 12: For our https data process we assumed it will be using Wi-Fi as that only made sense since the iPads are portable no other network is possible. It is also stated they are on the local network.

14: For the service application we knew the crew will be using this application so we assumed that the user will not have elevation. So, the application will be running as a standard user without elevation. The isolation of the app is within the app container as this is an app running on an iOS system. The service application should not be remoted into as the crew will have the ipads there as they work. It woold also not accept any network input so using that knowledge we stated the only input that it will accept is from the app itself. The app will be logging into by the member of the crew as the app must verify their certifications so the app will be using an authentication mechanism. Since the IOS device is handling the application the communication protocol work is mostly complete as apple handles a lot of that with the operating system so this will be used for this application.

15: For the human to use the system they must be authenticated so the source is authenticated as well as the destination as it is the physical device itself. So therefore, both must be authenticated.

16, 17, 18, and 19: For the information process between the application and the cache the device will control this restriction. So, both the source and destination will be authenticated as that deals with the inner workings of the operating system.

20 and 21: Both the caches within the iPad will be storing log data as discussed before we considered both service records and manuals/checklists to be log data. Since this data is internal to the device it will be encrypted by the operating system by default. This cache is to be updated so it will have written access in order to store any new data or overwrite existing data within the cache.

22: For the human interactor we set the authenticates itself to yes. We did this as we know for the crew to be verified, they will have to login to the system, so it knows who they are resulting in them being authenticated.

Threat Modeling Report

Created on 2020-09-30 4:07:44 PM

Threat Model Name:

Owner:

Reviewer:

Contributors:

Description:

Assumptions:

External Dependencies:

Threat Model Summary:

Not Started	0
Not Applicable	46
Needs Investigation	0
Mitigation Implemented	85
Total	131
Total Migrated	0

Diagram: Diagram 1

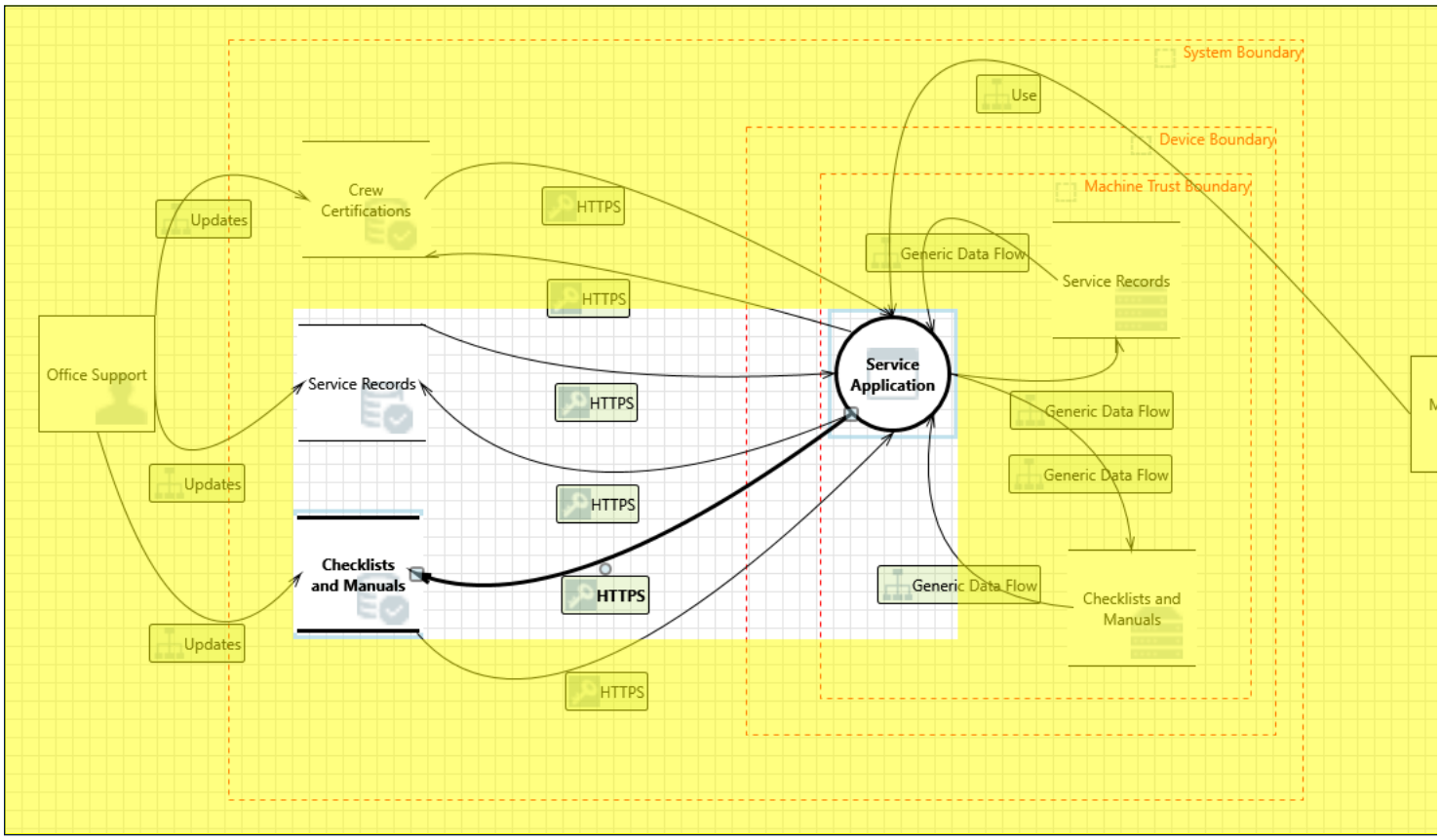
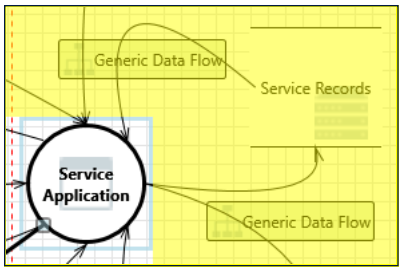


Diagram 1 Diagram Summary:

Not Started	0

Not Applicable	46
Needs Investigation	0
Mitigation Implemented	85
Total	131
Total Migrated	0

Interaction: Generic Data Flow



1. Spoofing of Source Data Store Service Records Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Spoofing
Description: Service Records may be spoofed by an attacker and this may lead to incorrect data delivered to Service Application. Consider using a standard authentication mechanism to identify the source data store.
Justification: Authentication with login and building security prevents unauthorized personnel.

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

Category: Information Disclosure
Description: Improper data protection of Service Records can allow an attacker to read information not intended for disclosure. Review authorization settings.
Justification: Assume maintenance crew are authorized and and restrict data on cache to be only what is needed to get the job done.

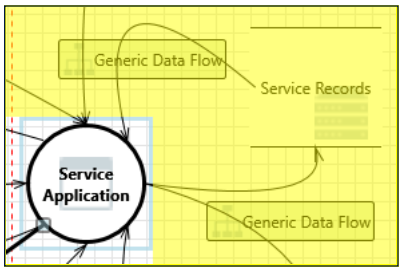
3. Risks from Logging Â [State: Not Applicable]Â [Priority: High]Â

Category: Tampering
Description: Log readers can come under attack via log files. Consider ways to canonicalize data in all logs. Implement a single reader for the logs, if possible, in order to reduce attack surface area. Be sure to understand and document log file elements which come from untrusted sources.
Justification: No applicable. Logs only are given one way from the service application to the database and are auto generated by the application. We assume the application is secure as access to application is regulated.

4. Authenticated Data Flow Compromised Â [State: Not Applicable]Â [Priority: High]Â

Category: Tampering
Description: An attacker can read or modify data transmitted over an authenticated dataflow.
Justification: We Assume anyone with access to the iPad is authorized.

Interaction: Generic Data Flow



5. Potential Excessive Resource Consumption for Service Application or Service Records Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Denial Of Service
Description: Does Service Application or Service Records 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: Explicit steps are used to ensure there is no deadlock. Read and write are done seperate and in sequential order. Timeouts are set.

6. Spoofing of Destination Data Store Service Records Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Spoofing
Description: Service Records may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of Service Records. Consider using a standard authentication mechanism to identify the destination data store.
Justification: Authentication with login and building security prevents unauthorized personnel.

7. Risks from Logging Â [State: Not Applicable]Â [Priority: High]Â

Category: Tampering
Description: Log readers can come under attack via log files. Consider ways to canonicalize data in all logs. Implement a single reader for the logs, if possible, in order to reduce attack surface area. Be sure to understand and document log file elements which come from untrusted sources.
Justification: No applicable. Logs only are given one way from the service application to the database and are auto generated by the application. We assume the application is secure as access to application is regulated.

8. Lower Trusted Subject Updates Logs Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation
Description: If you have trust levels, is anyone other outside of the highest trust level allowed to log? Letting everyone write to your logs can lead to repudiation problems. Only allow trusted code to log.
Justification: write access is only given to authorized and authenticated users via login though the service application.

9. Data Logs from an Unknown Source Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation
Description: Do you accept logs from unknown or weakly authenticated users or systems? Identify and authenticate the source of the logs before accepting them.
Justification: All logs must be authenticated by login before they can be sent. In addition, users of the service application are assumed authorized via building security.

10. Insufficient Auditing Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation
Description: Does the log capture enough data to understand what happened in the past? Do your logs capture enough data to understand an incident after the fact? Is such capture lightweight enough to be left on all the time? Do you have enough data to deal with repudiation claims? Make sure you log sufficient and appropriate data to handle a repudiation claims. You might want to talk to an audit expert as well as a privacy expert about your choice of data.
Justification: Logs capture both the user information such as name, employee ID, authorization level as well as general information such as time, date and captures, specific information such as what was done, etc.

11. Potential Weak Protections for Audit Data Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation
Description: Consider what happens when the audit mechanism comes under attack, including attempts to destroy the logs, or attack log analysis programs. Ensure access to the log is through a reference monitor, which controls read and write separately. Document what filters, if any, readers can rely on, or writers should expect
Justification: maitenance crew are assumed authorized and Service Application cannot modify contents in cache only read from and store logs.

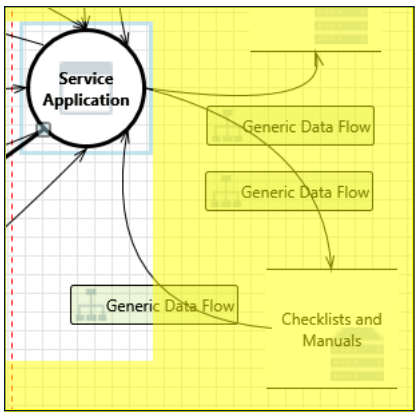
12. Authorization Bypass Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Information Disclosure
Description: Can you access Service Records 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: Use of encryption to prevent edititing.

13. Authenticated Data Flow Compromised Â [State: Not Applicable]Â [Priority: High]Â

Category: Tampering
Description: An attacker can read or modify data transmitted over an authenticated dataflow.
Justification: We Assume anyone with access to the iPad is authorized.

Interaction: Generic Data Flow



14. Spoofing of Destination Data Store Checklists and Manuals Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Spoofing

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

Justification: Maintenance crew are assumed to be authorized however authentication protocols are also used.

15. Potential Excessive Resource Consumption for Service Application or Checklists and Manuals Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Denial Of Service

Description: Does Service Application or Checklists and Manuals 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: Explicit steps are used to ensure there is no deadlock. Read and write are done seperate and in sequential order. Timeouts are set.

16. Risks from Logging Â [State: Not Applicable]Â [Priority: High]Â

Category: Tampering

Description: Log readers can come under attack via log files. Consider ways to canonicalize data in all logs. Implement a single reader for the logs, if possible, in order to reduce attack surface area. Be sure to understand and document log file elements which come from untrusted sources.

Justification: No applicable. Logs only are given one way from the service application to the database and are auto generated by the application. We assume the application is secure as access to application is regulated.

17. Lower Trusted Subject Updates Logs Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: If you have trust levels, is anyone other outside of the highest trust level allowed to log? Letting everyone write to your logs can lead to repudiation problems. Only allow trusted code to log.

Justification: write access is only given to authorized and authenticated users via login though the service application.

18. Data Logs from an Unknown Source Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: Do you accept logs from unknown or weakly authenticated users or systems? Identify and authenticate the source of the logs before accepting them.

Justification: All logs must be authenticated by login before they can be sent. In addition, users of the service application are assumed authorized via building security.

19. Insufficient Auditing Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: Does the log capture enough data to understand what happened in the past? Do your logs capture enough data to understand an incident after the fact? Is such capture lightweight enough to be left on all the time? Do you have enough data to deal with repudiation claims? Make sure you log sufficient and appropriate data to handle a repudiation claims. You might want to talk to an audit expert as well as a privacy expert about your choice of data.

Justification: Logs capture both the user information such as name, employee ID, authorization level as well as general information such as time, date and captures, specific information such as what was done, etc.

20. Potential Weak Protections for Audit Data Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: Consider what happens when the audit mechanism comes under attack, including attempts to destroy the logs, or attack log analysis programs. Ensure access to the log is through a reference monitor, which controls read and write separately. Document what filters, if any, readers can rely on, or writers should expect

Justification: maitenance crew are assumed authorized and Service Application cannot modify contents in cache only read from and store logs.

21. Authorization Bypass Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Information Disclosure

Description: Can you access Checklists and Manuals 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: Use of encryption to prevent edititing.

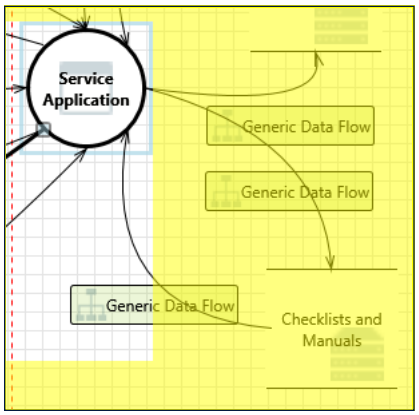
22. Authenticated Data Flow Compromised Â [State: Not Applicable]Â [Priority: High]Â

Category: Tampering

Description: An attacker can read or modify data transmitted over an authenticated dataflow.

Justification: We Assume anyone with access to the iPad is authorized.

Interaction: Generic Data Flow



23. Spoofing of Source Data Store Checklists and Manuals Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Spoofing

Description: Checklists and Manuals may be spoofed by an attacker and this may lead to incorrect data delivered to Service Application. Consider using a standard authentication mechanism to identify the source data store.

Justification: Maintenance crew are assumed to be authorized however authentication protocols are also used.

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

Category: Information Disclosure

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

Justification: Assume maintenance crew are authorized and and restrict data on cache to be only what is needed to get the job done.

25. Risks from Logging Â [State: Not Applicable]Â [Priority: High]Â

Category: Tampering

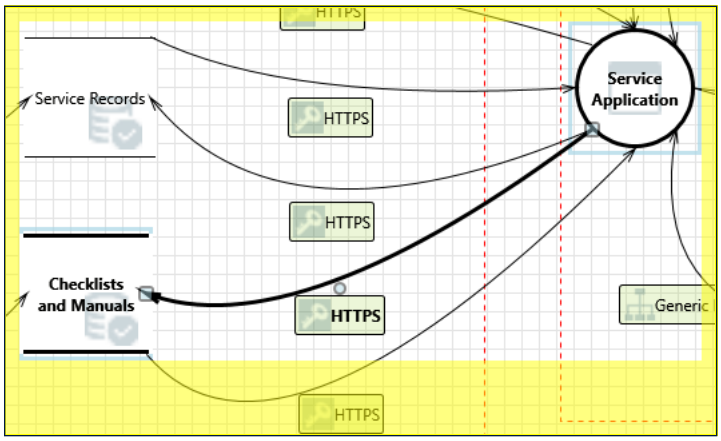
Description: Log readers can come under attack via log files. Consider ways to canonicalize data in all logs. Implement a single reader for the logs, if possible, in order to reduce attack surface area. Be sure to understand and document log file elements which come from untrusted sources.

Justification: No applicable. Logs only are given one way from the service application to the database and are auto generated by the application. We assume the application is secure as access to application is regulated.

26. Authenticated Data Flow Compromised Â [State: Not Applicable]Â [Priority: High]Â

Category: Tampering
Description: An attacker can read or modify data transmitted over an authenticated dataflow.
Justification: We Assume anyone with access to the iPad is authorized.

Interaction: HTTPS



27. Spoofing of Source Data Store Checklists and Manuals Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Spoofing
Description: Checklists and Manuals may be spoofed by an attacker and this may lead to incorrect data delivered to Service Application. Consider using a standard authentication mechanism to identify the source data store.
Justification: Both the database and service app use authentication to communicate as well being on the same local network of which non authorized personnel do not have access to making spoofing hard.

28. Potential Data Repudiation by Service Application Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation
Description: Service Application 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: Office Support must verify and perform any updates to the Service Application and sign off therefore there is a log done on the database side.

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

Category: Information Disclosure
Description: Improper data protection of Checklists and Manuals can allow an attacker to read information not intended for disclosure. Review authorization settings.
Justification: Service Application does not have read access to the database. Information from database that is needed in the Service Application is moved and stored directly on the iPad by Office Support and Service application simply writes log data to the database.

30. Potential Process Crash or Stop for Service Application Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Denial Of Service
Description: Service Application crashes, halts, stops or runs slowly; in all cases violating an availability metric.
Justification: When the database needs to send information to the service application to update its cache, if the service application is crashed the Office Support staff overseeing the update will be physically availble to fix the application. By desgin updates do not need to be done in realtime and if the application is crashed they can simply work to fix the application or retire the device for any amount of time untill the application is fixed or use a direct wire if there are network issues.

31. Data Flow HTTPS Is Potentially Interrupted Â [State: Not Applicable]Â [Priority: High]Â

Category: Denial Of Service
Description: An external agent interrupts data flowing across a trust boundary in either direction.
Justification: We are on a local network and as such it is assumed that all those who have access are authorized employees. If a DoS is indeed detected it would be up to security personnel to eliminate the problem.

32. Data Store Inaccessible Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Denial Of Service

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

Justification: We are on a local network and as such it is assumed that all those who have access are authorized employees. If a DoS is indeed detected it would be up to security personnel to eliminate the problem.

33. Service Application May be Subject to Elevation of Privilege Using Remote Code Execution Â [State: Not Applicable]Â [Priority: High]Â

Category: Elevation Of Privilege

Description: Checklists and Manuals may be able to remotely execute code for Service Application.

Justification: Database is directly controlled by Office support of which all are assumed to be authorized administrators of Service Application. If one is unauthorized it is a building security issue not software.

34. Elevation by Changing the Execution Flow in Service Application Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Elevation Of Privilege

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

Justification: Service Application requires a login authentication to use.

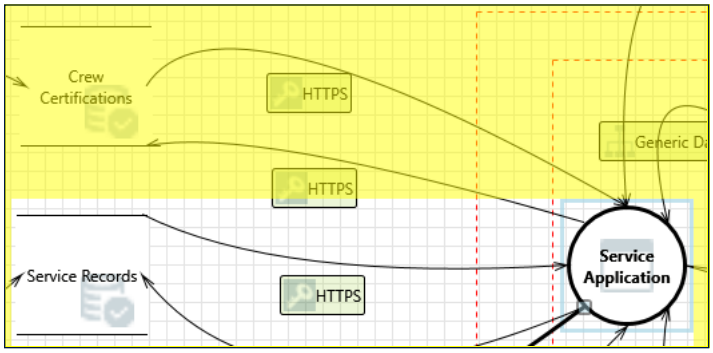
35. Risks from Logging Â [State: Not Applicable]Â [Priority: High]Â

Category: Tampering

Description: Log readers can come under attack via log files. Consider ways to canonicalize data in all logs. Implement a single reader for the logs, if possible, in order to reduce attack surface area. Be sure to understand and document log file elements which come from untrusted sources.

Justification: No applicable. Logs only are given one way from the service application to the database and are auto generated by the application. We assume the application is secure as access to application is regulated.

Interaction: HTTPS



36. Authorization Bypass Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Information Disclosure

Description: Can you access Crew Certifications 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: Local network require that tampering requries attackers to be physically in the building of which security will handle as well as use of encryption and firewalls.

37. Data Store Inaccessible Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Denial Of Service

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

Justification: We are on a local network and as such it is assumed that all those who have access are authorized employees. If a DoS is indeed detected it would be up to security personnel to eliminate the problem.

38. Data Flow HTTPS Is Potentially Interrupted Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Denial Of Service

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

Justification: We are on a local network and as such it is assumed that all those who have access are authorized employees. If a DoS is indeed detected it would be

up to security personnel to eliminate the problem.

39. Potential Excessive Resource Consumption for Service Application or Crew Certifications Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Denial Of Service

Description: Does Service Application or Crew Certifications 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: Service application only writes logs to database and 2 way communication only happens during updates performed by office support of which are not done in realtime preventing deadlocking.

40. Data Store Denies Crew Certifications Potentially Writing Data Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: Crew Certifications 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: Service Application writes to a log on the database. Log is write only and not readable or editable by the service application.

41. The Crew Certifications Data Store Could Be Corrupted Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Tampering

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

Justification: Data flow is encrypted and checked via standard error correction prcedures such as checksums etc.

42. Potential SQL Injection Vulnerability for Crew Certifications Â [State: Not Applicable]Â [Priority: High]Â

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: Service application only sends log files to database and cannot access the database directly or send it commands, database does not read logs sent.

43. Spoofing of Destination Data Store Crew Certifications Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Spoofing

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

Justification: Authentication with login and building security prevents unauthorized personnel.

44. Weak Credential Storage Â [State: Not Applicable]Â [Priority: High]Â

Category: Information Disclosure

Description: Credentials held at the server are often disclosed or tampered with and credentials stored on the client are often stolen. For server side, consider storing a salted hash of the credentials instead of storing the credentials themselves. If this is not possible due to business requirements, be sure to encrypt the credentials before storage, using an SDL-approved mechanism. For client side, if storing credentials is required, encrypt them and protect the data store in which they're stored

Justification: Service Application does not have the ability to edit or read data from the database.

45. Risks from Logging Â [State: Not Applicable]Â [Priority: High]Â

Category: Tampering

Description: Log readers can come under attack via log files. Consider ways to canonicalize data in all logs. Implement a single reader for the logs, if possible, in order to reduce attack surface area. Be sure to understand and document log file elements which come from untrusted sources.

Justification: No applicable. Logs only are given one way from the service application to the database and are auto generated by the application. We assume the application is secure as access to application is regulated.

46. Lower Trusted Subject Updates Logs Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: If you have trust levels, is anyone other outside of the highest trust level allowed to log? Letting everyone write to your logs can lead to repudiation problems. Only allow trusted code to log.

Justification: write access is only given to authorized and authenticated users via login though the service application.

47. Data Logs from an Unknown Source Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: Do you accept logs from unknown or weakly authenticated users or systems? Identify and authenticate the source of the logs before accepting them.

Justification: All logs must be authenticated by login before they can be sent. In addition, users of the service application are assumed authorized via building security.

48. Insufficient Auditing Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: Does the log capture enough data to understand what happened in the past? Do your logs capture enough data to understand an incident after the fact? Is such capture lightweight enough to be left on all the time? Do you have enough data to deal with repudiation claims? Make sure you log sufficient and appropriate data to handle a repudiation claims. You might want to talk to an audit expert as well as a privacy expert about your choice of data.

Justification: Logs capture both the user information such as name, employee ID, authorization level as well as general information such as time, date and captures, specific information such as what was done, etc.

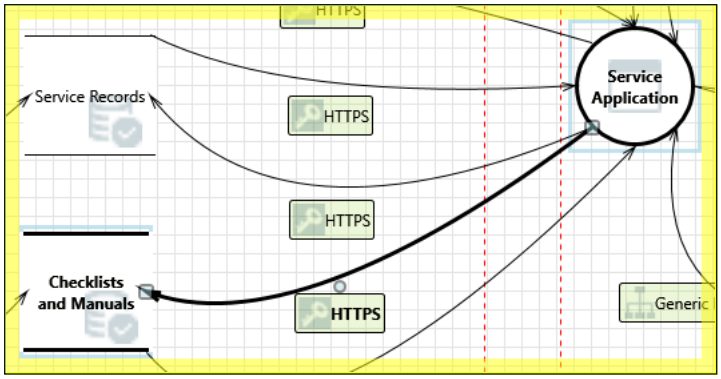
49. Potential Weak Protections for Audit Data Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: Consider what happens when the audit mechanism comes under attack, including attempts to destroy the logs, or attack log analysis programs. Ensure access to the log is through a reference monitor, which controls read and write separately. Document what filters, if any, readers can rely on, or writers should expect

Justification: Service Application only has write access to log data of what the crew did and cannot edit the other contents of the database, or read any data from the database.

Interaction: HTTPS



50. Spoofing of Destination Data Store Checklists and Manuals Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Spoofing

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

Justification: Both the database and service app use authentication to communicate as well being on the same local network of which non authorized personnel do not have access to making spoofing hard.

51. Potential SQL Injection Vulnerability for Checklists and Manuals Â [State: Not Applicable]Â [Priority: High]Â

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: Service application only sends log files to database and cannot access the database directly or send it commands, database does not read logs sent.

52. The Checklists and Manuals Data Store Could Be Corrupted Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Tampering

Description: Data flowing across HTTPS may be tampered with by an attacker. This may lead to corruption of Checklists and Manuals. Ensure the integrity of the data flow to the data store.

Justification: Data flow is encrypted and checked via standard error correction prcedures such as checksums etc.

53. Data Store Denies Checklists and Manuals Potentially Writing Data Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: Checklists and Manuals 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: Service Application writes to a log on the database. Log is write only and not readable or editable by the service application.

54. Potential Excessive Resource Consumption for Service Application or Checklists and Manuals Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Denial Of Service

Description: Does Service Application or Checklists and Manuals 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: Service application only writes logs to database and 2 way communication only happens during updates performed by office support of which are not done in realtime preventing deadlocking.

55. Data Flow HTTPS Is Potentially Interrupted Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Denial Of Service

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

Justification: We are on a local network and as such it is assumed that all those who have access are authorized employees. If a DoS is indeed detected it would be up to security personnel to eliminate the problem.

56. Data Store Inaccessible Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Denial Of Service

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

Justification: We are on a local network and as such it is assumed that all those who have access are authorized employees. If a DoS is indeed detected it would be up to security personnel to eliminate the problem.

57. Authorization Bypass Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Information Disclosure

Description: Can you access Checklists and Manuals 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: Local network require that tampering requries attackers to be physically in the building of which security will handle as well as use of encryption and firewalls.

58. Risks from Logging Â [State: Not Applicable]Â [Priority: High]Â

Category: Tampering

Description: Log readers can come under attack via log files. Consider ways to canonicalize data in all logs. Implement a single reader for the logs, if possible, in order to reduce attack surface area. Be sure to understand and document log file elements which come from untrusted sources.

Justification: No applicable. Logs only are given one way from the service application to the database and are auto generated by the application. We assume the application is secure as access to application is regulated.

59. Lower Trusted Subject Updates Logs Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: If you have trust levels, is anyone other outside of the highest trust level allowed to log? Letting everyone write to your logs can lead to repudiation problems. Only allow trusted code to log.

Justification: write access is only given to authorized and authenticated users via login though the service application.

60. Data Logs from an Unknown Source Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: Do you accept logs from unknown or weakly authenticated users or systems? Identify and authenticate the source of the logs before accepting them.

Justification: All logs must be authenticated by login before they can be sent. In addition, users of the service application are assumed authorized via building

security.

61. Insufficient Auditing Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: Does the log capture enough data to understand what happened in the past? Do your logs capture enough data to understand an incident after the fact? Is such capture lightweight enough to be left on all the time? Do you have enough data to deal with repudiation claims? Make sure you log sufficient and appropriate data to handle a repudiation claims. You might want to talk to an audit expert as well as a privacy expert about your choice of data.

Justification: Logs capture both the user information such as name, employee ID, authorization level as well as general information such as time, date and captures, specific information such as what was done, etc.

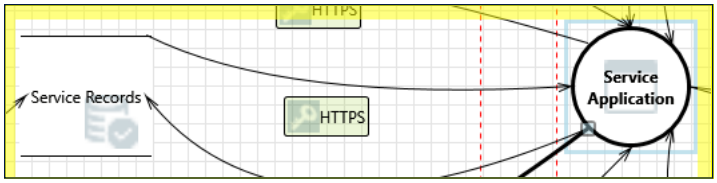
62. Potential Weak Protections for Audit Data Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: Consider what happens when the audit mechanism comes under attack, including attempts to destroy the logs, or attack log analysis programs. Ensure access to the log is through a reference monitor, which controls read and write separately. Document what filters, if any, readers can rely on, or writers should expect

Justification: Service Application only has write access to log data of what the crew did and cannot edit the other contents of the database, or read any data from the database.

Interaction: HTTPS



63. Spoofing of Source Data Store Service Records Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Spoofing

Description: Service Records may be spoofed by an attacker and this may lead to incorrect data delivered to Service Application. Consider using a standard authentication mechanism to identify the source data store.

Justification: Authentication with login and building security prevents unauthorized personnel.

64. Potential Data Repudiation by Service Application Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: Service Application 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: Office Support must verify and perform any udates to the Service Application and sign off therefore there is a log done on the database side.

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

Category: Information Disclosure

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

Justification: Service Application does not have read access to the database. Information from database that is needed in the Service Application is moved and stored directly on the iPad by Office Support and Service application simply writes log data to the database.

66. Potential Process Crash or Stop for Service Application Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Denial Of Service

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

Justification: When the database needs to send information to the service application to update its cache, if the service application is crashed the Office Support staff overseeing the update will be physically availble to fix the application. By desgin updates do not need to be done in realtime and if the application is crashed they can simply work to fix the application or retire the device for any amount of time untill the application is fixed or use a direct wire if there are network issues.

67. Data Flow HTTPS Is Potentially Interrupted Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Denial Of Service

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

Justification: We are on a local network and as such it is assumed that all those who have access are authorized employees. If a DoS is indeed detected it would be up to security personnel to eliminate the problem.

68. Data Store Inaccessible Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Denial Of Service

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

Justification: We are on a local network and as such it is assumed that all those who have access are authorized employees. If a DoS is indeed detected it would be up to security personnel to eliminate the problem.

69. Service Application May be Subject to Elevation of Privilege Using Remote Code Execution Â [State: Not Applicable]Â [Priority: High]Â

Category: Elevation Of Privilege

Description: Service Records may be able to remotely execute code for Service Application.

Justification: Database is directly controlled by Office support of which all are assumed to be authorized administrators of Service Application. If one is unauthorized it is a building security issue not software.

70. Elevation by Changing the Execution Flow in Service Application Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Elevation Of Privilege

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

Justification: Service Application requires a login authentication to use.

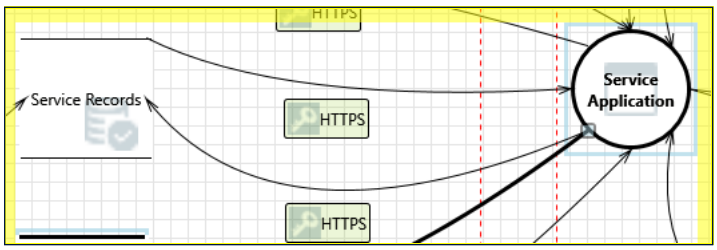
71. Risks from Logging Â [State: Not Applicable]Â [Priority: High]Â

Category: Tampering

Description: Log readers can come under attack via log files. Consider ways to canonicalize data in all logs. Implement a single reader for the logs, if possible, in order to reduce attack surface area. Be sure to understand and document log file elements which come from untrusted sources.

Justification: No applicable. Logs only are given one way from the service application to the database and are auto generated by the application. We assume the application is secure as access to application is regulated.

Interaction: HTTPS



72. Authorization Bypass Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Information Disclosure

Description: Can you access Service Records 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: Local network require that tampering requries attackers to be physically in the building of which security will handle as well as use of encryption and firewalls.

73. Spoofing of Destination Data Store Service Records Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Spoofing

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

Justification: Authentication with login and building security prevents unauthorized personnel.

74. Potential SQL Injection Vulnerability for Service Records Â [State: Not Applicable]Â [Priority: High]Â

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: Service application only sends log files to database and cannot access the database directly or send it commands, database does not read logs sent.

75. The Service Records Data Store Could Be Corrupted Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Tampering

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

Justification: Data flow is encrypted and checked via standard error correction prcedures such as checksums etc.

76. Data Store Denies Service Records Potentially Writing Data Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: Service Records 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: Service Application writes to a log on the database. Log is write only and not readable or editable by the service application.

77. Potential Excessive Resource Consumption for Service Application or Service Records Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Denial Of Service

Description: Does Service Application or Service Records 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: Service application only writes logs to database and 2 way communication only happens during updates performed by office support of which are not done in realtime preventing deadlocking.

78. Data Flow HTTPS Is Potentially Interrupted Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Denial Of Service

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

Justification: We are on a local network and as such it is assumed that all those who have access are authorized employees. If a DoS is indeed detected it would be up to security personnel to eliminate the problem.

79. Data Store Inaccessible Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Denial Of Service

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

Justification: We are on a local network and as such it is assumed that all those who have access are authorized employees. If a DoS is indeed detected it would be up to security personnel to eliminate the problem.

80. Risks from Logging Â [State: Not Applicable]Â [Priority: High]Â

Category: Tampering

Description: Log readers can come under attack via log files. Consider ways to canonicalize data in all logs. Implement a single reader for the logs, if possible, in order to reduce attack surface area. Be sure to understand and document log file elements which come from untrusted sources.

Justification: No applicable. Logs only are given one way from the service application to the database and are auto generated by the application. We assume the application is secure as access to application is regulated.

81. Lower Trusted Subject Updates Logs Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: If you have trust levels, is anyone other outside of the highest trust level allowed to log? Letting everyone write to your logs can lead to repudiation problems. Only allow trusted code to log.

Justification: write access is only given to authorized and authenticated users via login though the service application.

82. Data Logs from an Unknown Source Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: Do you accept logs from unknown or weakly authenticated users or systems? Identify and authenticate the source of the logs before accepting them.

Justification: All logs must be authenticated by login before they can be sent. In addition, users of the service application are assumed authorized via building

security.

83. Insufficient Auditing Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: Does the log capture enough data to understand what happened in the past? Do your logs capture enough data to understand an incident after the fact? Is such capture lightweight enough to be left on all the time? Do you have enough data to deal with repudiation claims? Make sure you log sufficient and appropriate data to handle a repudiation claims. You might want to talk to an audit expert as well as a privacy expert about your choice of data.

Justification: Logs capture both the user information such as name, employee ID, authorization level as well as general information such as time, date and captures, specific information such as what was done, etc.

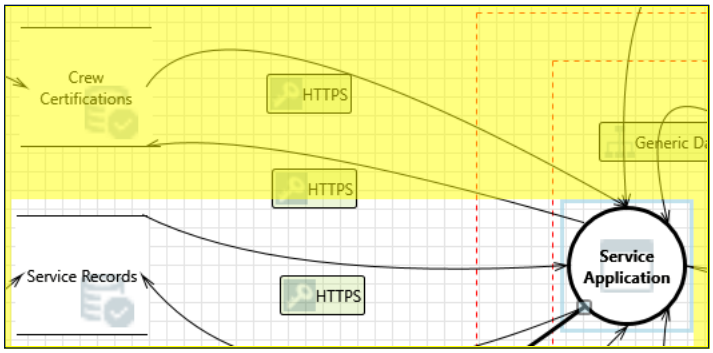
84. Potential Weak Protections for Audit Data Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: Consider what happens when the audit mechanism comes under attack, including attempts to destroy the logs, or attack log analysis programs. Ensure access to the log is through a reference monitor, which controls read and write separately. Document what filters, if any, readers can rely on, or writers should expect

Justification: Service Application only has write access to log data of what the crew did and cannot edit the other contents of the database, or read any data from the database.

Interaction: HTTPS



85. Spoofing of Source Data Store Crew Certifications Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Spoofing

Description: Crew Certifications may be spoofed by an attacker and this may lead to incorrect data delivered to Service Application. Consider using a standard authentication mechanism to identify the source data store.

Justification: Authentication with login and building security prevents unauthorized personnel.

86. Potential Data Repudiation by Service Application Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: Service Application 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: Office Support must verify and perform any udates to the Service Application and sign off therefore there is a log done on the database side.

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

Category: Information Disclosure

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

Justification: Service Application does not have read access to the database. Information from database that is needed in the Service Application is moved and stored directly on the iPad by Office Support and Service application simply writes log data to the database.

88. Potential Process Crash or Stop for Service Application Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Denial Of Service

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

Justification: When the database needs to send information to the service application to update its cache, if the service application is crashed the Office Support staff overseeing the update will be physically availble to fix the application. By desgin updates do not need to be done in realtime and if the

application is crashed they can simply work to fix the application or retire the device for any amount of time untill the application is fixed or use a direct wire if there are network issues.

89. Data Flow HTTPS Is Potentially Interrupted Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Denial Of Service
Description: An external agent interrupts data flowing across a trust boundary in either direction.
Justification: We are on a local network and as such it is assumed that all those who have access are authorized employees. If a DoS is indeed detected it would be up to security personnel to eliminate the problem.

90. Data Store Inaccessible Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Denial Of Service
Description: An external agent prevents access to a data store on the other side of the trust boundary.
Justification: We are on a local network and as such it is assumed that all those who have access are authorized employees. If a DoS is indeed detected it would be up to security personnel to eliminate the problem.

91. Service Application May be Subject to Elevation of Privilege Using Remote Code Execution Â [State: Not Applicable]Â [Priority: High]Â

Category: Elevation Of Privilege
Description: Crew Certifications may be able to remotely execute code for Service Application.
Justification: Database is directly controlled by Office support of which all are assumed to be authorized administrators of Service Application. If one is unauthorized it is a building security issue not software.

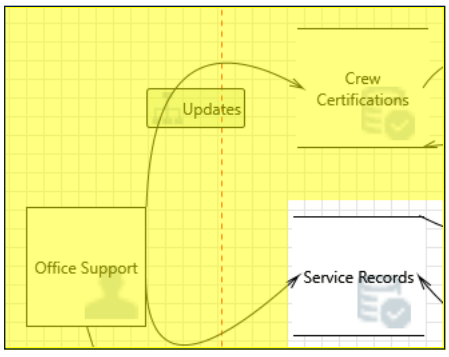
92. Elevation by Changing the Execution Flow in Service Application Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Elevation Of Privilege
Description: An attacker may pass data into Service Application in order to change the flow of program execution within Service Application to the attacker's choosing.
Justification: Service Application requires a login authentication to use.

93. Risks from Logging Â [State: Not Applicable]Â [Priority: High]Â

Category: Tampering
Description: Log readers can come under attack via log files. Consider ways to canonicalize data in all logs. Implement a single reader for the logs, if possible, in order to reduce attack surface area. Be sure to understand and document log file elements which come from untrusted sources.
Justification: No applicable. Logs only are given one way from the service application to the database and are auto generated by the application. We assume the application is secure as access to application is regulated.

Interaction: Updates



94. Spoofing of Destination Data Store Crew Certifications Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Spoofing
Description: Crew Certifications may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of Crew Certifications. Consider using a standard authentication mechanism to identify the destination data store.
Justification: Authentication with login and building security prevents unauthorized personnel.

95. Possible SQL Injection Vulnerability for Crew Certifications Â [State: Not Applicable]Â [Priority: High]Â

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: Assume that all Office Support is authorized and as such will not attack database.

96. The Crew Certifications Data Store Could Be Corrupted Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Tampering

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

Justification: Office Support assumed to be authenticated and connected to databases by wire.

97. Data Store Denies Crew Certifications Potentially Writing Data Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: Crew Certifications 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: There will be a local log at the Office Support end that logs who wrote or otherwise accessed the databases. This log will be held offline to prevent remote tampering and protection for this log will be achieved by physical security such that no unauthorized personnel may enter the office that the Office Support staff reside and we are assuming all staff who enter and thus can write to the log is authorized.

98. Data Flow Updates Is Potentially Interrupted Â [State: Not Applicable]Â [Priority: High]Â

Category: Denial Of Service

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

Justification: We Assume anyone with access to Office Support is authorized.

99. Data Store Inaccessible Â [State: Not Applicable]Â [Priority: High]Â

Category: Denial Of Service

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

Justification: We Assume anyone with access to Office Support is authorized.

100. Authenticated Data Flow Compromised Â [State: Not Applicable]Â [Priority: High]Â

Category: Tampering

Description: An attacker can read or modify data transmitted over an authenticated dataflow.

Justification: We Assume anyone with access to Office Support is authorized.

101. Lower Trusted Subject Updates Logs Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation

Description: If you have trust levels, is anyone other outside of the highest trust level allowed to log? Letting everyone write to your logs can lead to repudiation problems. Only allow trusted code to log.

Justification: Office Support assumed to be authorized. is also authenticated with login.

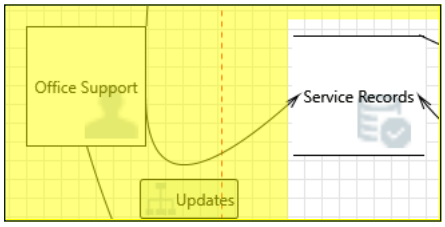
102. Data Logs from an Unknown Source Â [State: Not Applicable]Â [Priority: High]Â

Category: Repudiation

Description: Do you accept logs from unknown or weakly authenticated users or systems? Identify and authenticate the source of the logs before accepting them.

Justification: Office Support is assumed authorized.

Interaction: Updates



103. Spoofing of Destination Data Store Service Records Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Spoofing
Description: Service Records may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of Service Records. Consider using a standard authentication mechanism to identify the destination data store.
Justification: Authentication with login and building security prevents unauthorized personnel.

104. Possible SQL Injection Vulnerability for Service Records Â [State: Not Applicable]Â [Priority: High]Â

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: Assume that all Office Support is authorized and as such will not attack database.

105. The Service Records Data Store Could Be Corrupted Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Tampering
Description: Data flowing across Updates may be tampered with by an attacker. This may lead to corruption of Service Records. Ensure the integrity of the data flow to the data store.
Justification: Office Support assumed to be authenticated and connected to databases by wire.

106. Data Store Denies Service Records Potentially Writing Data Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation
Description: Service Records 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: There will be a local log at the Office Support end that logs who wrote or otherwise accessed the databases. This log will be held offline to prevent remote tampering and protection for this log will be achieved by physical security such that no unauthorized personnel may enter the office that the Office Support staff reside and we are assuming all staff who enter and thus can write to the log is authorized.

107. Data Flow Updates Is Potentially Interrupted Â [State: Not Applicable]Â [Priority: High]Â

Category: Denial Of Service
Description: An external agent interrupts data flowing across a trust boundary in either direction.
Justification: We Assume anyone with access to Office Support is authorized.

108. Data Store Inaccessible Â [State: Not Applicable]Â [Priority: High]Â

Category: Denial Of Service
Description: An external agent prevents access to a data store on the other side of the trust boundary.
Justification: We Assume anyone with access to Office Support is authorized.

109. Authenticated Data Flow Compromised Â [State: Not Applicable]Â [Priority: High]Â

Category: Tampering
Description: An attacker can read or modify data transmitted over an authenticated dataflow.
Justification: We Assume anyone with access to Office Support is authorized.

110. Lower Trusted Subject Updates Logs Â [State: Mitigation Implemented]Â [Priority: High]Â

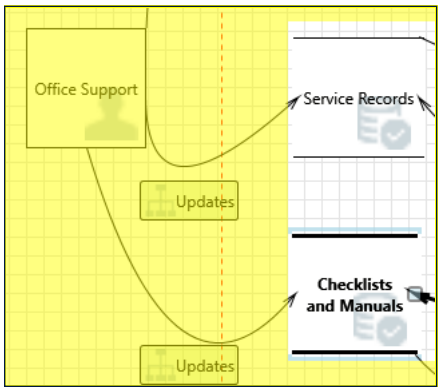
Category: Repudiation
Description: If you have trust levels, is anyone other outside of the highest trust level allowed to log? Letting everyone write to your logs can lead to repudiation problems. Only allow trusted code to log.

Justification: Office Support assumed to be authorized. is also authenticated with login.

111. Data Logs from an Unknown Source Â [State: Not Applicable]Â [Priority: High]Â

Category: Repudiation
Description: Do you accept logs from unknown or weakly authenticated users or systems? Identify and authenticate the source of the logs before accepting them.
Justification: Office Support is assumed authorized.

Interaction: Updates



112. Spoofing of Destination Data Store Checklists and Manuals Â [State: Not Applicable]Â [Priority: High]Â

Category: Spoofing
Description: Checklists and Manuals may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of Checklists and Manuals. Consider using a standard authentication mechanism to identify the destination data store.
Justification: Office Support are assumed to be authenticated. Access to database from that end are physical wire.

113. Possible SQL Injection Vulnerability for Checklists and Manuals Â [State: Not Applicable]Â [Priority: High]Â

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: Assume that all Office Support is authorized and as such will not attack database.

114. The Checklists and Manuals Data Store Could Be Corrupted Â [State: Not Applicable]Â [Priority: High]Â

Category: Tampering
Description: Data flowing across Updates may be tampered with by an attacker. This may lead to corruption of Checklists and Manuals. Ensure the integrity of the data flow to the data store.
Justification: Office Support assumed to be authenticated and connected to databases by wire.

115. Data Store Denies Checklists and Manuals Potentially Writing Data Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation
Description: Checklists and Manuals 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: There will be a local log at the Office Support end that logs who wrote or otherwise accessed the databases. This log will be held offline to prevent remote tampering and protection for this log will be achieved by physical security such that no unauthorized personnel may enter the office that the Office Support staff reside and we are assuming all staff who enter and thus can write to the log is authorized.

116. Data Flow Updates Is Potentially Interrupted Â [State: Not Applicable]Â [Priority: High]Â

Category: Denial Of Service
Description: An external agent interrupts data flowing across a trust boundary in either direction.
Justification: We Assume anyone with access to Office Support is authorized.

117. Data Store Inaccessible Â [State: Not Applicable]Â [Priority: High]Â

Category: Denial Of Service
Description: An external agent prevents access to a data store on the other side of the trust boundary.
Justification: We Assume anyone with access to Office Support is authorized.

118. Authenticated Data Flow Compromised Â [State: Not Applicable]Â [Priority: High]Â

Category: Tampering
Description: An attacker can read or modify data transmitted over an authenticated dataflow.
Justification: We Assume anyone with access to Office Support is authorized.

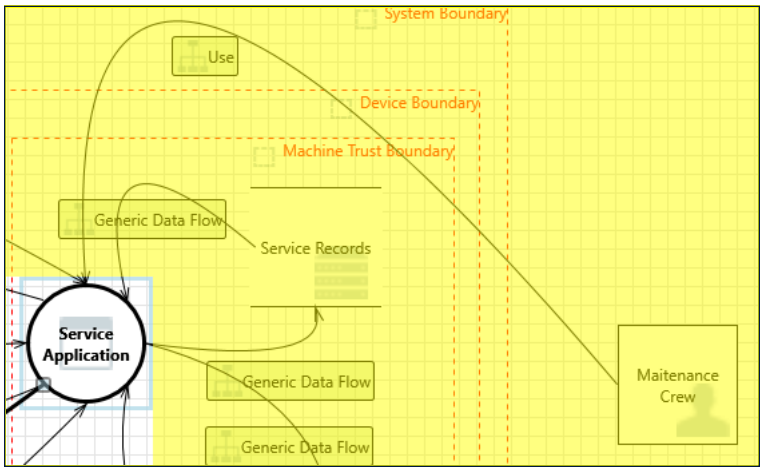
119. Lower Trusted Subject Updates Logs Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Repudiation
Description: If you have trust levels, is anyone other outside of the highest trust level allowed to log? Letting everyone write to your logs can lead to repudiation problems. Only allow trusted code to log.
Justification: Office Support assumed to be authorized. is also authenticated with login.

120. Data Logs from an Unknown Source Â [State: Not Applicable]Â [Priority: High]Â

Category: Repudiation
Description: Do you accept logs from unknown or weakly authenticated users or systems? Identify and authenticate the source of the logs before accepting them.
Justification: Office Support is assumed authorized.

Interaction: Use



121. Spoofing the Service Application Process Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Spoofing
Description: Service Application may be spoofed by an attacker and this may lead to information disclosure by Maitenance Crew. Consider using a standard authentication mechanism to identify the destination process.
Justification: Authentication with login and building security prevents unauthorized personnel.

122. Authenticated Data Flow Compromised Â [State: Not Applicable]Â [Priority: High]Â

Category: Tampering
Description: An attacker can read or modify data transmitted over an authenticated dataflow.
Justification: We Assume anyone with access to Maintenance Crew is authorized.

123. Potential Lack of Input Validation for Service Application Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Tampering
Description: Data flowing across Use may be tampered with by an attacker. This may lead to a denial of service attack against Service Application or an elevation of privilege attack against Service Application or an information disclosure by Service Application. 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: Physical use by human. Maintanence crew assumed authorized.

124. Potential Data Repudiation by Service Application Â [State: Not Applicable]Â [Priority: High]Â

Category: Repudiation
Description: Service Application 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: Physical user using device.

125. Data Flow Sniffing Â [State: Not Applicable]Â [Priority: High]Â

Category: Information Disclosure
Description: Data flowing across Use 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: User is physically using the iPad.

126. Potential Process Crash or Stop for Service Application Â [State: Not Applicable]Â [Priority: High]Â

Category: Denial Of Service
Description: Service Application crashes, halts, stops or runs slowly; in all cases violating an availability metric.
Justification: User will call Office Suport. They will either fix or send a new iPad.

127. Data Flow Use Is Potentially Interrupted Â [State: Not Applicable]Â [Priority: High]Â

Category: Denial Of Service
Description: An external agent interrupts data flowing across a trust boundary in either direction.
Justification: We Assume anyone with access to Maintenance Crew is authorized.

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

Category: Elevation Of Privilege
Description: Service Application may be able to impersonate the context of Maitenance Crew in order to gain additional privilege.
Justification: Maintenance crew are assumed to be authorized by building security.

129. Service Application May be Subject to Elevation of Privilege Using Remote Code Execution Â [State: Not Applicable]Â [Priority: High]Â

Category: Elevation Of Privilege
Description: Maitenance Crew may be able to remotely execute code for Service Application.
Justification: Database is directly controlled by Office support of which all are assumed to be authorized administrators of Service Application. If one is unauthorized it is a building security issue not software.

130. Elevation by Changing the Execution Flow in Service Application Â [State: Mitigation Implemented]Â [Priority: High]Â

Category: Elevation Of Privilege
Description: An attacker may pass data into Service Application in order to change the flow of program execution within Service Application to the attacker's choosing.
Justification: Service Application requires a login authentication to use.

131. Cross Site Request Forgery Â [State: Not Applicable]Â [Priority: High]Â

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: <no mitigation provided>