Threat Modeling Report

Created on 12/3/2018 7:39:46 PM

Threat Model Name:

Owner:

Reviewer:

Contributors:

Description:

Assumptions:

External Dependencies:

Threat Model Summary:

Not Started	43
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	43
Total Migrated	0

Diagram: Diagram 1

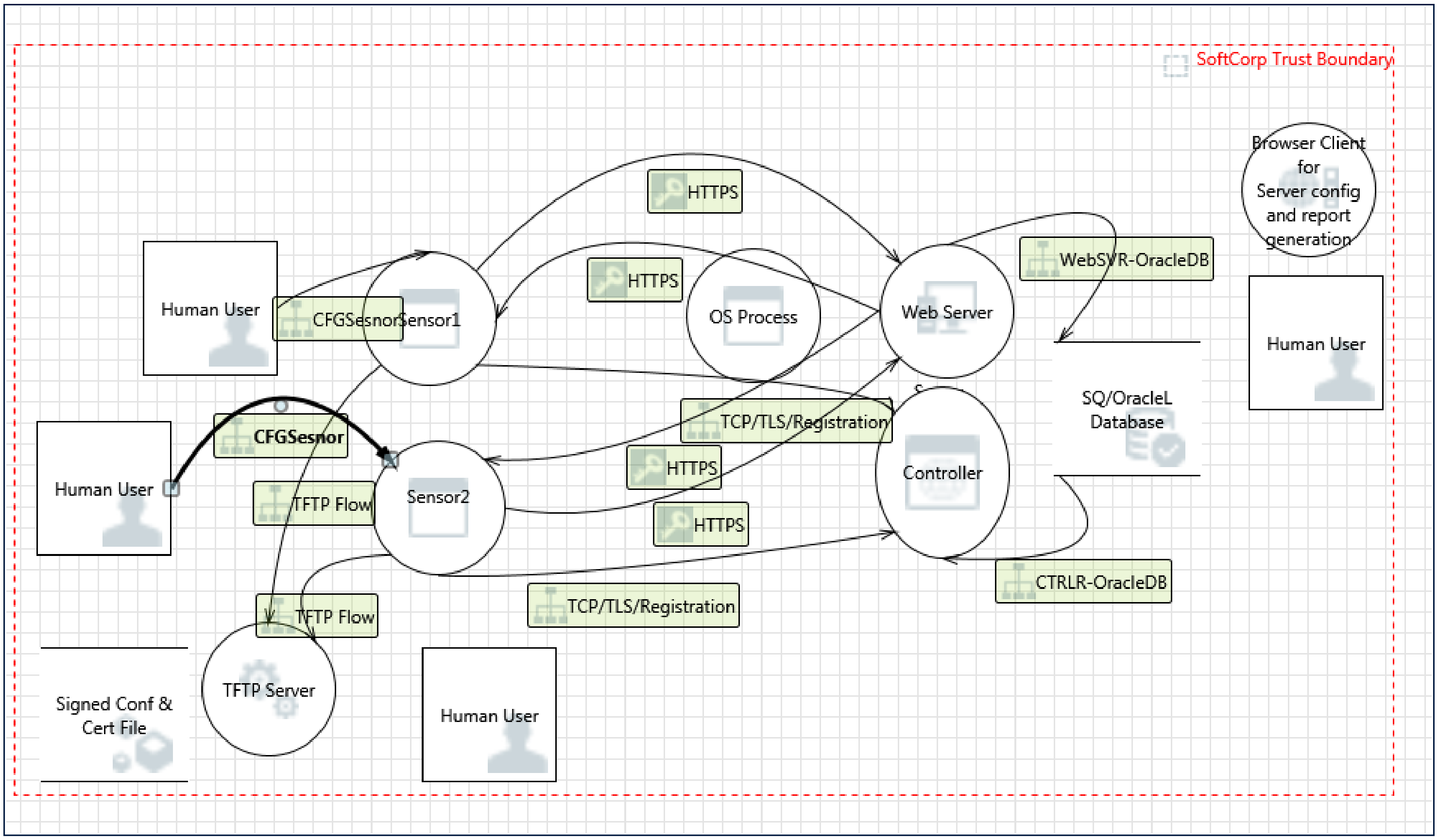
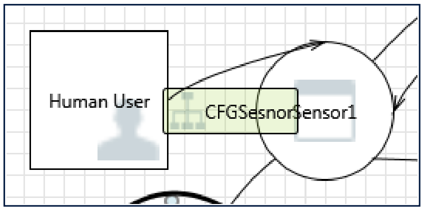


Diagram 1 Diagram Summary:

Not Started	43
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	43
Total Migrated	0

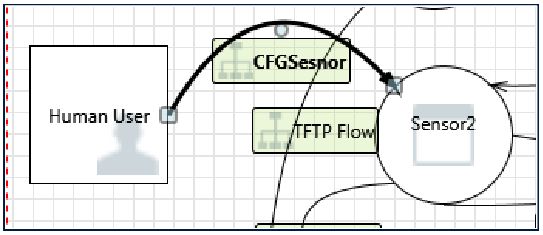
Interaction: CFGSesnor



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

Category: Elevation Of Privilege
Description: Sensor1 may be able to impersonate the context of Human User in order to gain additional privilege.
Justification: <no mitigation provided>

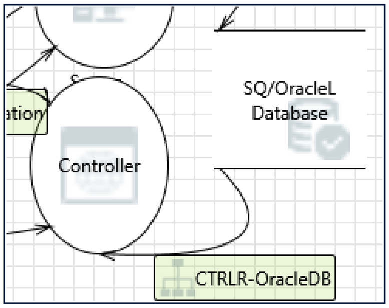
Interaction: CFGSesnor



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

Category: Elevation Of Privilege
Description: Sensor2 may be able to impersonate the context of Human User in order to gain additional privilege.
Justification: <no mitigation provided>

Interaction: CTRLR-OracleDB



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

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

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

Category: Information Disclosure

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

Justification: <no mitigation provided>

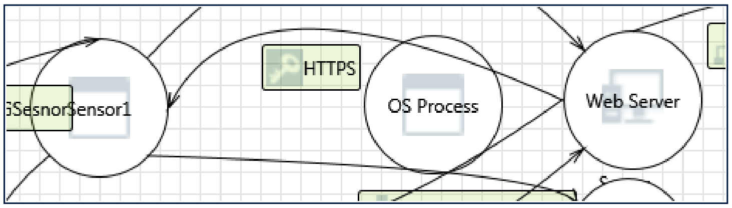
5. Spoofing of Source Data Store SQ/OracleL Database [State: Not Started] [Priority: High]

Category: Spoofing

Description: SQ/OracleL Database may be spoofed by an attacker and this may lead to incorrect data delivered to Controller. Consider using a standard authentication mechanism to identify the source data store.

Justification: <no mitigation provided>

Interaction: HTTPS



6. Collision Attacks [State: Not Started] [Priority: High]

Category: Tampering

Description: Attackers who can send a series of packets or messages may be able to overlap data. For example, packet 1 may be 100 bytes starting at offset 0. Packet 2 may be 100 bytes starting at offset 25. Packet 2 will overwrite 75 bytes of packet 1. Ensure you reassemble data before filtering it, and ensure you explicitly handle these sorts of cases.

Justification: <no mitigation provided>

7. Replay Attacks [State: Not Started] [Priority: High]

Category: Tampering

Description: Packets or messages without sequence numbers or timestamps can be captured and replayed in a wide variety of ways. Implement or utilize an existing communication protocol that supports anti-replay techniques (investigate sequence numbers before timers) and strong integrity.

Justification: <no mitigation provided>

8. Web Server Process Memory Tampered [State: Not Started] [Priority: High]

Category: Tampering

Description: If Web Server is given access to memory, such as shared memory or pointers, or is given the ability to control what Sensor1 executes (for example, passing back a function pointer.), then Web Server can tamper with Sensor1. Consider if the function could work with less access to memory, such as passing data rather than pointers. Copy in data provided, and then validate it.

Justification: <no mitigation provided>

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

Category: Elevation Of Privilege

Description: Sensor1 may be able to impersonate the context of Web Server in order to gain additional privilege.

Justification: <no mitigation provided>

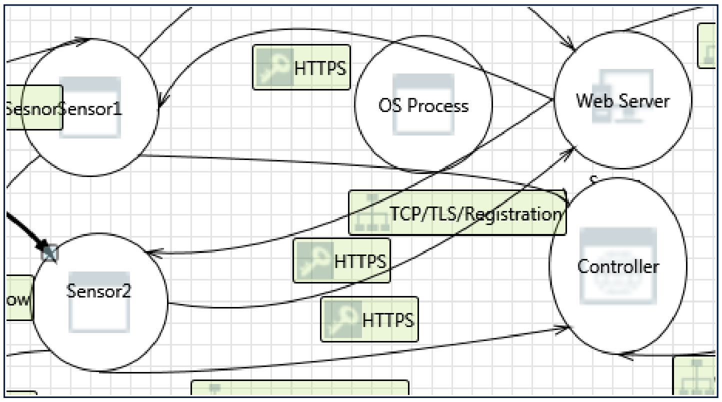
10. Weak Authentication Scheme [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: Custom authentication schemes are susceptible to common weaknesses such as weak credential change management, credential equivalence, easily guessable credentials, null credentials, downgrade authentication or a weak credential change management system. Consider the impact and potential mitigations for your custom authentication scheme.

Justification: <no mitigation provided>

Interaction: HTTPS



11. Replay Attacks [State: Not Started] [Priority: High]

Category: Tampering

Description: Packets or messages without sequence numbers or timestamps can be captured and replayed in a wide variety of ways. Implement or utilize an existing communication protocol that supports anti-replay techniques (investigate sequence numbers before timers) and strong integrity.

Justification: <no mitigation provided>

12. Web Server Process Memory Tampered [State: Not Started] [Priority: High]

Category: Tampering

Description: If Web Server is given access to memory, such as shared memory or pointers, or is given the ability to control what Sensor2 executes (for example, passing back a function pointer.), then Web Server can tamper with Sensor2 . Consider if the function could work with less access to memory, such as passing data rather than pointers. Copy in data provided, and then validate it.

Justification: <no mitigation provided>

13. Collision Attacks [State: Not Started] [Priority: High]

Category: Tampering

Description: Attackers who can send a series of packets or messages may be able to overlap data. For example, packet 1 may be 100 bytes starting at offset 0. Packet 2 may be 100 bytes starting at offset 25. Packet 2 will overwrite 75 bytes of packet 1. Ensure you reassemble data before filtering it, and ensure you explicitly handle these sorts of cases.

Justification: <no mitigation provided>

14. Weak Authentication Scheme [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: Custom authentication schemes are susceptible to common weaknesses such as weak credential change management, credential equivalence, easily guessable credentials, null credentials, downgrade authentication or a weak credential change management system. Consider the impact and potential mitigations for your custom authentication scheme.

Justification: <no mitigation provided>

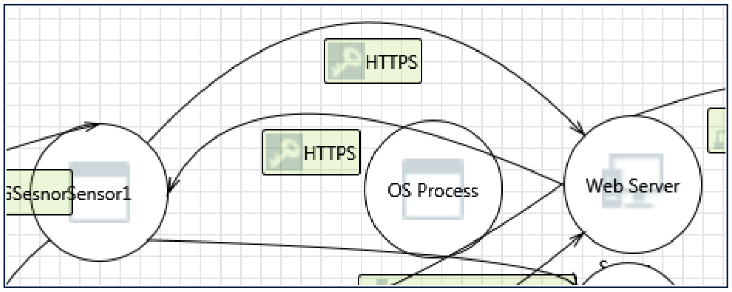
15. Elevation Using Impersonation [State: Not Started] [Priority: High]

Category: Elevation Of Privilege

Description: Sensor2 may be able to impersonate the context of Web Server in order to gain additional privilege.

Justification: <no mitigation provided>

Interaction: HTTPS



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

Category: Elevation Of Privilege
Description: Web Server may be able to impersonate the context of Sensor1 in order to gain additional privilege.
Justification: <no mitigation provided>

17. Collision Attacks [State: Not Started] [Priority: High]

Category: Tampering
Description: Attackers who can send a series of packets or messages may be able to overlap data. For example, packet 1 may be 100 bytes starting at offset 0. Packet 2 may be 100 bytes starting at offset 25. Packet 2 will overwrite 75 bytes of packet 1. Ensure you reassemble data before filtering it, and ensure you explicitly handle these sorts of cases.
Justification: <no mitigation provided>

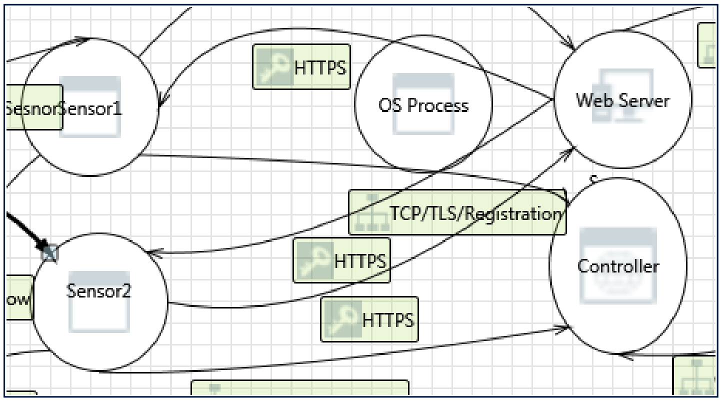
18. Replay Attacks [State: Not Started] [Priority: High]

Category: Tampering
Description: Packets or messages without sequence numbers or timestamps can be captured and replayed in a wide variety of ways. Implement or utilize an existing communication protocol that supports anti-replay techniques (investigate sequence numbers before timers) and strong integrity.
Justification: <no mitigation provided>

19. Weak Authentication Scheme [State: Not Started] [Priority: High]

Category: Information Disclosure
Description: Custom authentication schemes are susceptible to common weaknesses such as weak credential change management, credential equivalence, easily guessable credentials, null credentials, downgrade authentication or a weak credential change management system. Consider the impact and potential mitigations for your custom authentication scheme.
Justification: <no mitigation provided>

Interaction: HTTPS



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

Category: Elevation Of Privilege
Description: Web Server may be able to impersonate the context of Sensor2 in order to gain additional privilege.
Justification: <no mitigation provided>

21. Weak Authentication Scheme   [State: Not Started] [Priority: High]

Category: Information Disclosure
Description: Custom authentication schemes are susceptible to common weaknesses such as weak credential change management, credential equivalence, easily guessable credentials, null credentials, downgrade authentication or a weak credential change management system. Consider the impact and potential mitigations for your custom authentication scheme.
Justification: <no mitigation provided>

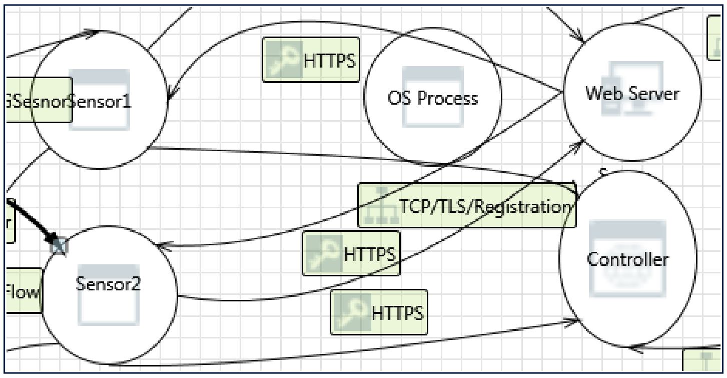
22. Collision Attacks   [State: Not Started] [Priority: High]

Category: Tampering
Description: Attackers who can send a series of packets or messages may be able to overlap data. For example, packet 1 may be 100 bytes starting at offset 0. Packet 2 may be 100 bytes starting at offset 25. Packet 2 will overwrite 75 bytes of packet 1. Ensure you reassemble data before filtering it, and ensure you explicitly handle these sorts of cases.
Justification: <no mitigation provided>

23. Replay Attacks   [State: Not Started] [Priority: High]

Category: Tampering
Description: Packets or messages without sequence numbers or timestamps can be captured and replayed in a wide variety of ways. Implement or utilize an existing communication protocol that supports anti-replay techniques (investigate sequence numbers before timers) and strong integrity.
Justification: <no mitigation provided>

Interaction: TCP/TLS/Registration



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

Category: Elevation Of Privilege
Description: Controller may be able to impersonate the context of Sensor1 in order to gain additional privilege.
Justification: <no mitigation provided>

25. Replay Attacks   [State: Not Started] [Priority: High]

Category: Tampering
Description: Packets or messages without sequence numbers or timestamps can be captured and replayed in a wide variety of ways. Implement or utilize an existing communication protocol that supports anti-replay techniques (investigate sequence numbers before timers) and strong integrity.
Justification: <no mitigation provided>

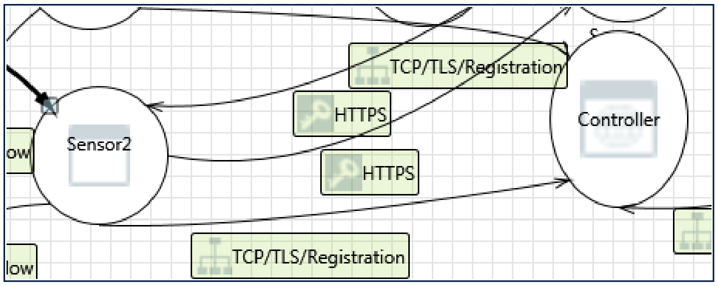
26. Weak Authentication Scheme   [State: Not Started] [Priority: High]

Category: Information Disclosure
Description: Custom authentication schemes are susceptible to common weaknesses such as weak credential change management, credential equivalence, easily guessable credentials, null credentials, downgrade authentication or a weak credential change management system. Consider the impact and potential mitigations for your custom authentication scheme.
Justification: <no mitigation provided>

27. Collision Attacks [State: Not Started] [Priority: High]

Category: Tampering
Description: Attackers who can send a series of packets or messages may be able to overlap data. For example, packet 1 may be 100 bytes starting at offset 0. Packet 2 may be 100 bytes starting at offset 25. Packet 2 will overwrite 75 bytes of packet 1. Ensure you reassemble data before filtering it, and ensure you explicitly handle these sorts of cases.
Justification: <no mitigation provided>

Interaction: TCP/TLS/Registration



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

Category: Elevation Of Privilege
Description: Controller may be able to impersonate the context of Sensor2 in order to gain additional privilege.
Justification: <no mitigation provided>

29. Weak Authentication Scheme [State: Not Started] [Priority: High]

Category: Information Disclosure
Description: Custom authentication schemes are susceptible to common weaknesses such as weak credential change management, credential equivalence, easily guessable credentials, null credentials, downgrade authentication or a weak credential change management system. Consider the impact and potential mitigations for your custom authentication scheme.
Justification: <no mitigation provided>

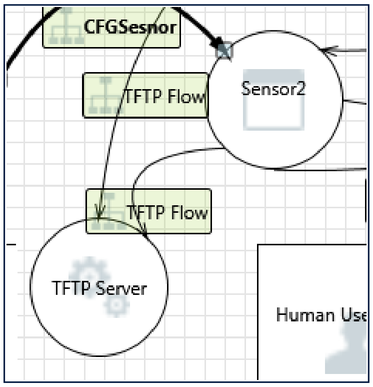
30. Collision Attacks [State: Not Started] [Priority: High]

Category: Tampering
Description: Attackers who can send a series of packets or messages may be able to overlap data. For example, packet 1 may be 100 bytes starting at offset 0. Packet 2 may be 100 bytes starting at offset 25. Packet 2 will overwrite 75 bytes of packet 1. Ensure you reassemble data before filtering it, and ensure you explicitly handle these sorts of cases.
Justification: <no mitigation provided>

31. Replay Attacks [State: Not Started] [Priority: High]

Category: Tampering
Description: Packets or messages without sequence numbers or timestamps can be captured and replayed in a wide variety of ways. Implement or utilize an existing communication protocol that supports anti-replay techniques (investigate sequence numbers before timers) and strong integrity.
Justification: <no mitigation provided>

Interaction: TFTP Flow



32. Weak Authentication Scheme [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: Custom authentication schemes are susceptible to common weaknesses such as weak credential change management, credential equivalence, easily guessable credentials, null credentials, downgrade authentication or a weak credential change management system. Consider the impact and potential mitigations for your custom authentication scheme.

Justification: <no mitigation provided>

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

Category: Elevation Of Privilege

Description: TFTP Server may be able to impersonate the context of Sensor2 in order to gain additional privilege.

Justification: <no mitigation provided>

34. Collision Attacks [State: Not Started] [Priority: High]

Category: Tampering

Description: Attackers who can send a series of packets or messages may be able to overlap data. For example, packet 1 may be 100 bytes starting at offset 0. Packet 2 may be 100 bytes starting at offset 25. Packet 2 will overwrite 75 bytes of packet 1. Ensure you reassemble data before filtering it, and ensure you explicitly handle these sorts of cases.

Justification: <no mitigation provided>

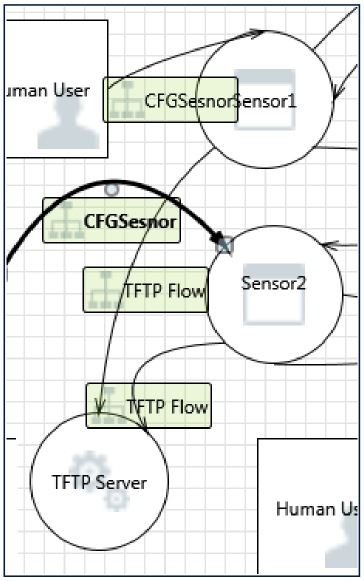
35. Replay Attacks [State: Not Started] [Priority: High]

Category: Tampering

Description: Packets or messages without sequence numbers or timestamps can be captured and replayed in a wide variety of ways. Implement or utilize an existing communication protocol that supports anti-replay techniques (investigate sequence numbers before timers) and strong integrity.

Justification: <no mitigation provided>

Interaction: TFTP Flow



36. Collision Attacks [State: Not Started] [Priority: High]

Category: Tampering

Description: Attackers who can send a series of packets or messages may be able to overlap data. For example, packet 1 may be 100 bytes starting at offset 0. Packet 2 may be 100 bytes starting at offset 25. Packet 2 will overwrite 75 bytes of packet 1. Ensure you reassemble data before filtering it, and ensure you explicitly handle these sorts of cases.

Justification: <no mitigation provided>

37. Replay Attacks [State: Not Started] [Priority: High]

Category: Tampering

Description: Packets or messages without sequence numbers or timestamps can be captured and replayed in a wide variety of ways. Implement or utilize an existing communication protocol that supports anti-replay techniques (investigate sequence numbers before timers) and strong integrity.

Justification: <no mitigation provided>

38. Weak Authentication Scheme [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: Custom authentication schemes are susceptible to common weaknesses such as weak credential change management, credential equivalence, easily guessable credentials, null credentials, downgrade authentication or a weak credential change management system. Consider the impact and potential mitigations for your custom authentication scheme.

Justification: <no mitigation provided>

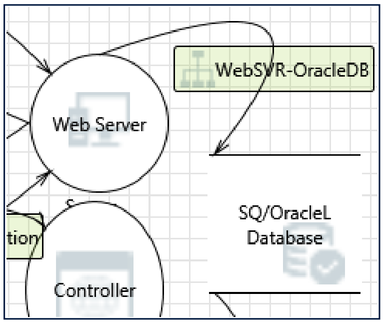
39. Elevation Using Impersonation [State: Not Started] [Priority: High]

Category: Elevation Of Privilege

Description: TFTP Server may be able to impersonate the context of Sensor1 in order to gain additional privilege.

Justification: <no mitigation provided>

Interaction: WebSVR-OracleDB



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

Category: Information Disclosure

Description: Can you access SQ/OracleL Database and bypass the permissions for the object? For example by editing the files directly with a hex editor, or reaching it via filesharing? Ensure that your program is the only one that can access the data, and that all other subjects have to use your interface.

Justification: <no mitigation provided>

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

Category: Denial Of Service

Description: Does Web Server or SQ/OracleL 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: <no mitigation provided>

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

Category: Tampering

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

Justification: <no mitigation provided>

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

Category: Spoofing

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

Justification: <no mitigation provided>