Threat Modeling Report

Created on 11/15/2021 8:30:02 PM

Threat Model Name:

Owner:

Reviewer:

Contributors:

Description:

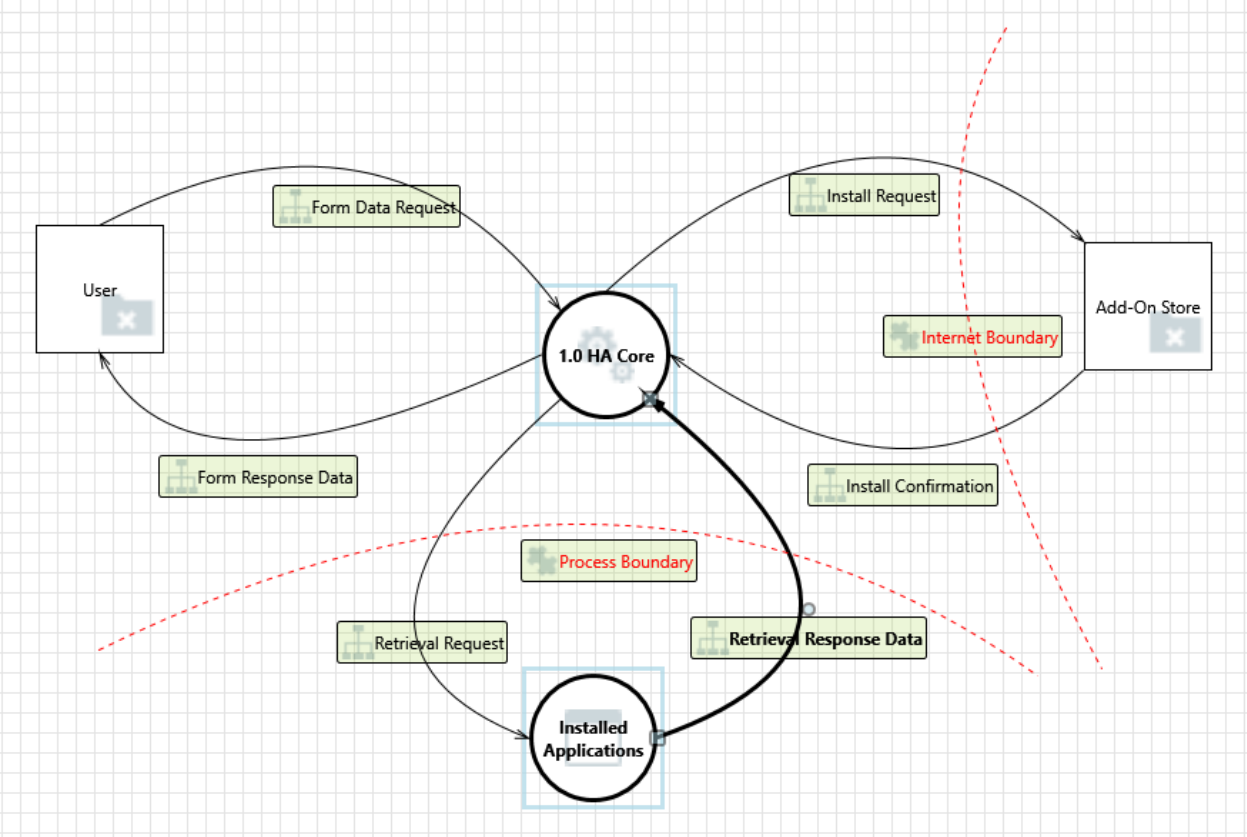
Assumptions:

External Dependencies:

Threat Model Summary:

Not Started	0
Not Applicable	28
Needs Investigation	3
Mitigation Implemented	6
Total	37
Total Migrated	0

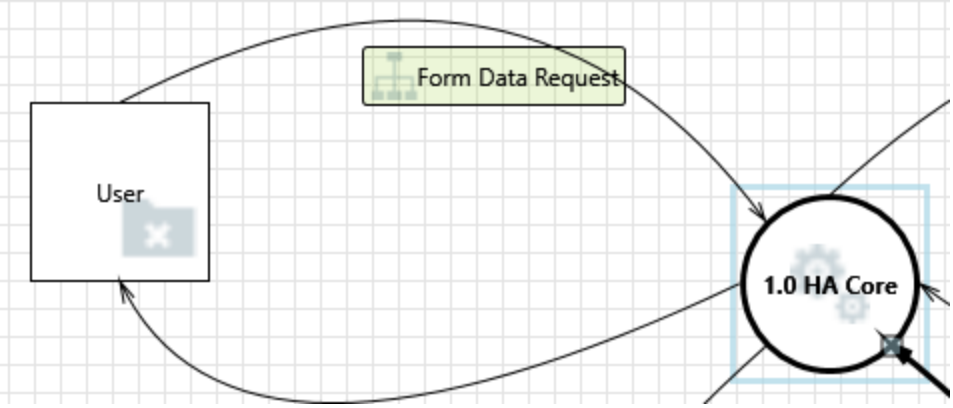
Diagram: DFD Level 1.1 Add-Ons



DFD Level 1.1 Add-Ons Diagram Summary:

Not Started	0
Not Applicable	28
Needs Investigation	3
Mitigation Implemented	6
Total	37
Total Migrated	0

Interaction: Form Data Request



1. Spoofing the User External Entity [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: User may be spoofed by an attacker and this may lead to unauthorized access to 1.0 HA Core. Consider using a standard authentication mechanism to identify the external entity.

Justification: User is authenticated using a standard username/password combination. Timeout counter. Authentication is available on HA https://www.home-assistant.io/docs/authentication.

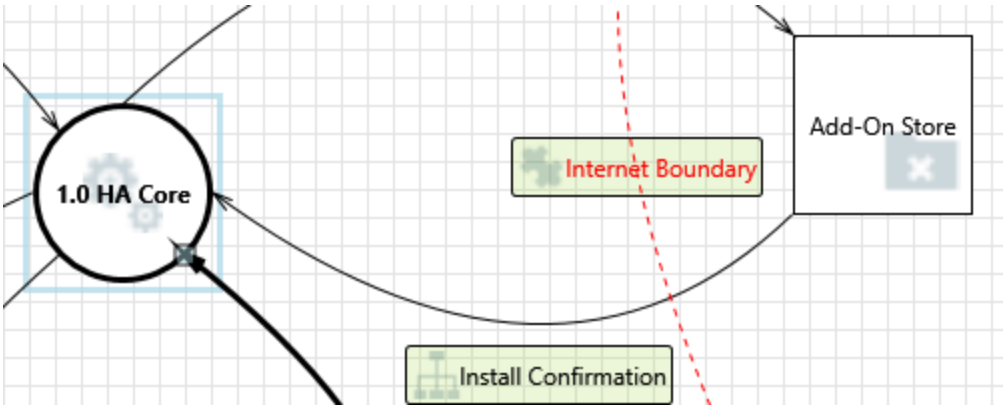
2. Elevation Using Impersonation [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

Description: 1.0 HA Core may be able to impersonate the context of User in order to gain additional privilege.

Justification: HA Core takes input from User, not the other way around in this instance. Not Applicable.

Interaction: Install Confirmation



3. Spoofing the 1.0 HA Core Process [State: Not Applicable] [Priority: High]

Category: Spoofing
Description: 1.0 HA Core may be spoofed by an attacker and this may lead to information disclosure by Add-On Store. Consider using a standard authentication mechanism to identify the destination process.
Justification: To spoof HA Core and access an installed application the attacker would already need admin rights on the machine that HA Core is running on. Not applicable.

4. Elevation Using Impersonation [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege
Description: 1.0 HA Core may be able to impersonate the context of Add-On Store in order to gain additional privilege.
Justification: HA Core takes input from Add-On Store, not the other way around in this instance. Not Applicable.

5. Spoofing the Add-On Store External Entity [State: Not Applicable] [Priority: High]

Category: Spoofing
Description: Add-On Store may be spoofed by an attacker and this may lead to unauthorized access to 1.0 HA Core. Consider using a standard authentication mechanism to identify the external entity.
Justification: <no mitigation provided>

6. Potential Lack of Input Validation for 1.0 HA Core [State: Needs Investigation] [Priority: High]

Category: Tampering
Description: Data flowing across Install Confirmation may be tampered with by an attacker. This may lead to a denial of service attack against 1.0 HA Core or an elevation of

privilege attack against 1.0 HA Core or an information disclosure by 1.0 HA Core. 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: Is there a checking process (hash checking or authentication of full download) from Add-On Store to HA Core? Could be that not all files download for the Add-On?

7. Potential Data Repudiation by 1.0 HA Core [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: 1.0 HA Core 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: Check for logging and audit. Data logging from both the core and IoT devices is available with the logger integration, https://www.home-assistant.io/integrations/logger.

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

Category: Information Disclosure

Description: Data flowing across Install Confirmation 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: Not sure if data from Add-On Store is encrypted while sending, would depend on connection type (HTTPS vs HTTP). Data Encryption is vital.

9. Potential Process Crash or Stop for 1.0 HA Core [State: Not Applicable] [Priority: High]

Category: Denial Of Service

Description: 1.0 HA Core crashes, halts, stops or runs slowly; in all cases violating an availability metric.

Justification: Add-Ons are only downloaded to HA Core when HA Core requests them.

10. Data Flow Install Confirmation Is Potentially Interrupted [State: Not Applicable] [Priority: High]

Category: Denial Of Service

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

Justification: Add-On would just not install to HA Core.

11. 1.0 HA Core May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

Description: Add-On Store may be able to remotely execute code for 1.0 HA Core.

Justification: Add-Ons only install when requested by HA Core. Maintain ACL that does not allow Add-ons to be automatically installed.

12. Elevation by Changing the Execution Flow in 1.0 HA Core [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: An attacker may pass data into 1.0 HA Core in order to change the flow of program execution within 1.0 HA Core to the attacker's choosing.

Justification: HA should not allow installed applications more than the least amount of privilege required.

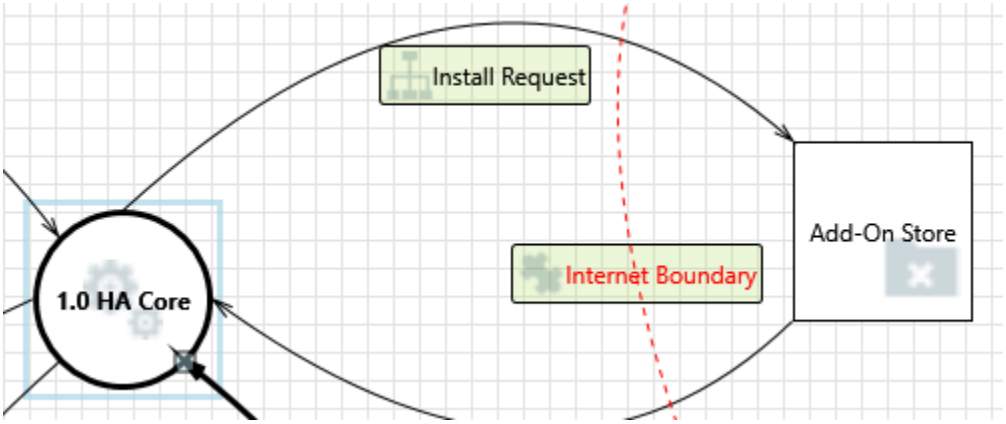
13. Cross Site Request Forgery [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

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

Justification: Add-On Store is accessable only though the HA Add-On Store page, this is within HA Core, and not on a browser so a CSRF attack would not be applicable.

Interaction: Install Request



14. Data Flow Install Request Is Potentially Interrupted [State: Not Applicable] [Priority: High]

Category: Denial Of Service

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

Justification: Add-On will just not install into HA Core. While annoying, it will not disclose any data from HA Core.

15. External Entity Add-On Store Potentially Denies Receiving Data [State: Not Applicable] [Priority: High]

Category: Repudiation

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

Justification: HA Core will just show that the Add-On was not downloaded. Not Applicable.

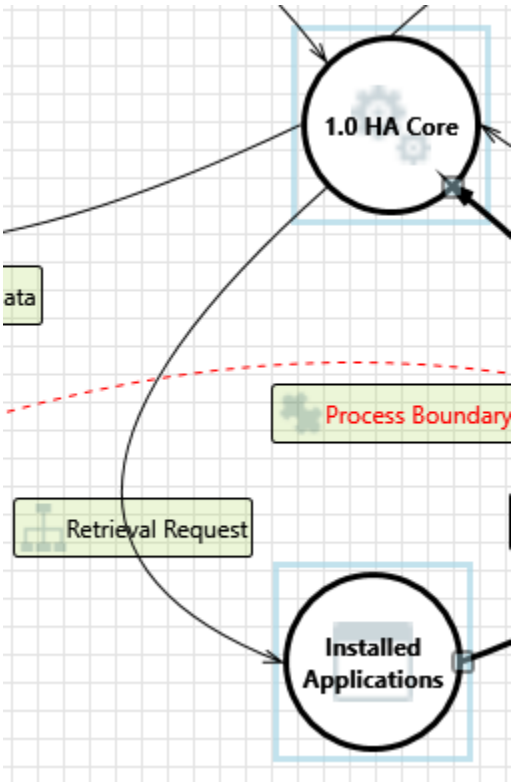
16. Spoofing of the Add-On Store External Destination Entity [State: Not Applicable] [Priority: High]

Category: Spoofing

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

Justification: Though add-ons can be malicious, the app store itself cannot be spoofed.

Interaction: Retrieval Request



17. Data Flow Retrieval Request Is Potentially Interrupted [State: Not Applicable] [Priority: High]

Category: Denial Of Service

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

Justification: To create a DoS attack from an installed application the attacker would already need admin rights on the machine that HA Core is running on. No Requirement to protect the installed applicaton process

18. Potential Process Crash or Stop for Installed Applications [State: Not Applicable] [Priority: High]

Category: Denial Of Service

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

Justification: To create a DoS attack from an installed application the attacker would already need admin rights on the machine that HA Core is running on. No Requirement to protect the installed applicaton process

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

Category: Information Disclosure

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

Justification: Check for encryption. Encryption can be set up with the Let's Encrypt integration.

20. Potential Data Repudiation by Installed Applications [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Installed Applications 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: Consider using auditing and logging. Data logging from both the core and IoT devices is available with the logger integration, https://www.home-assistant.io/integrations/logger.

21. Potential Lack of Input Validation for Installed Applications [State: Not Applicable] [Priority: High]

Category: Tampering

Description: Data flowing across Retrieval Request may be tampered with by an attacker. This may lead to a denial of service attack against Installed Applications or an elevation of privilege attack against Installed Applications or an information disclosure by Installed Applications. 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: To tamper with Retrival Request as it connects to an installed application the attacker would already need admin rights on the machine that HA Core is runnign on. Not applicable.

22. Spoofing the Installed Applications Process [State: Not Applicable] [Priority: High]

Category: Spoofing

Description: Installed Applications may be spoofed by an attacker and this may lead to information disclosure by 1.0 HA Core. Consider using a standard authentication mechanism to identify the destination process.

Justification: To spoof an installed application the attacker would already need admin rights on the machine that HA Core is running on. Not applicable.

23. Spoofing the 1.0 HA Core Process [State: Not Applicable] [Priority: High]

Category: Spoofing

Description: 1.0 HA Core may be spoofed by an attacker and this may lead to unauthorized access to Installed Applications. Consider using a standard authentication mechanism to identify the source process.

Justification: To spoof HA Core and access an installed application the attacker would already need admin rights on the machine that HA Core is running on. Not applicable.

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

Category: Elevation Of Privilege

Description: Installed Applications may be able to impersonate the context of 1.0 HA Core in order to gain additional privilege.

Justification: To gain additional privliage from an installed application the attacker would already need admin rights on the machine that HA Core is runnign on. Not applicable.

25. Installed Applications May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

Description: 1.0 HA Core may be able to remotely execute code for Installed Applications.

Justification: To remotely execute code from an installed application the attacker would already need admin rights on the machine that HA Core is runnign on. Not applicable.

26. Elevation by Changing the Execution Flow in Installed Applications [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

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

Justification: To change the process flow from an installed application the attacker would already need admin rights on the machine that HA Core is runnign on. Not applicable.

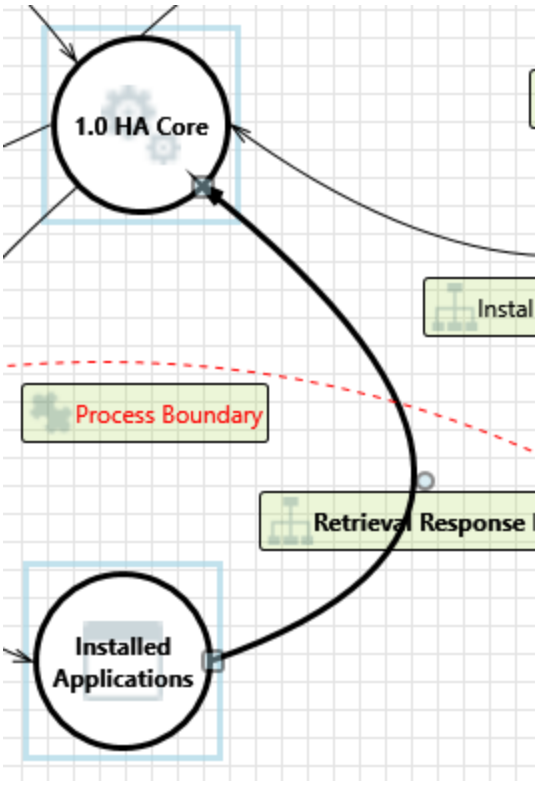
27. Cross Site Request Forgery [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

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

Justification: As installed applications and HA Core run on the same machine, a CSRF would not be possible as no browser is used. Not applicable.

Interaction: Retrieval Response Data



28. Elevation by Changing the Execution Flow in 1.0 HA Core [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

Description: An attacker may pass data into 1.0 HA Core in order to change the flow of program execution within 1.0 HA Core to the attacker's choosing.

Justification: To pass data from an installed application the attacker would already need admin rights on the machine that HA Core is running on. HA should not allow installed applications more than the least amount of privilege required.

29. 1.0 HA Core May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

Description: Installed Applications may be able to remotely execute code for 1.0 HA Core.

Justification: To remotely execute code from an installed application the attacker would already need admin rights on the machine that HA Core is running on. This is also a normal designed function of the software. HA should not allow installed applications more than the least amount of privilege required.

30. Elevation Using Impersonation [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

Description: 1.0 HA Core may be able to impersonate the context of Installed Applications in order to gain additional privilege.

Justification: To impersonate an installed application the attacker would already need admin rights on the machine that HA Core is running on. HA should not allow installed applications more than the least amount of privilege required.

31. Data Flow Retrieval Response Data Is Potentially Interrupted [State: Not Applicable] [Priority: High]

Category: Denial Of Service

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

Justification: To stop data flow from an installed application the attacker would already need admin rights on the machine that HA Core is running on. Prevent excess disk or CPU consumption.

32. Potential Process Crash or Stop for 1.0 HA Core [State: Not Applicable] [Priority: High]

Category: Denial Of Service

Description: 1.0 HA Core crashes, halts, stops or runs slowly; in all cases violating an availability metric.

Justification: To create a DoS attack from an installed application the attacker would already need admin rights on the machine that HA Core is running on. Prevent excess disk or CPU consumption.

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

Category: Information Disclosure

Description: Data flowing across Retrieval Response Data 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: check for encryption.

34. Potential Data Repudiation by 1.0 HA Core [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: 1.0 HA Core 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: Check for logging and audit. Data logging from both the core and IoT devices is available with the logger integration, https://www.home-assistant.io/integrations/logger.

35. Potential Lack of Input Validation for 1.0 HA Core [State: Needs Investigation] [Priority: High]

Category: Tampering

Description: Data flowing across Retrieval Response Data may be tampered with by an attacker. This may lead to a denial of service attack against 1.0 HA Core or an elevation of privilege attack against 1.0 HA Core or an information disclosure by 1.0 HA Core. 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: Verify that all input is verified for correctness using an approved list input validation approach.

36. Spoofing the 1.0 HA Core Process [State: Not Applicable] [Priority: High]

Category: Spoofing

Description: 1.0 HA Core may be spoofed by an attacker and this may lead to information disclosure by Installed Applications. Consider using a standard authentication mechanism to identify the destination process.

Justification: To spoof an already installed application the attacker would need admin rights to HA Core. Not Applicable.

37. Spoofing the Installed Applications Process [State: Not Applicable] [Priority: High]

Category: Spoofing

Description: Installed Applications may be spoofed by an attacker and this may lead to unauthorized access to 1.0 HA Core. Consider using a standard authentication mechanism to identify the source process.

Justification: To spoof an already installed application the attacker would need admin rights to HA Core. Not Applicable.