Threat Modeling Report

Created on 11/15/2020 12:59:10 PM

Threat Model Name:

Owner:

Reviewer:

Contributors:

Description:

Assumptions:

External Dependencies:

Notes:

Id	Note	Date	Added By
1		11/14/2020 8:48:13 PM	DESKTOP-LVH9R6U\Jose

Threat Model Summary:

Not Started	0
Not Applicable	6
Needs Investigation	0
Mitigation Implemented	26
Total	32
Total Migrated	0

Diagram: Diagram 1

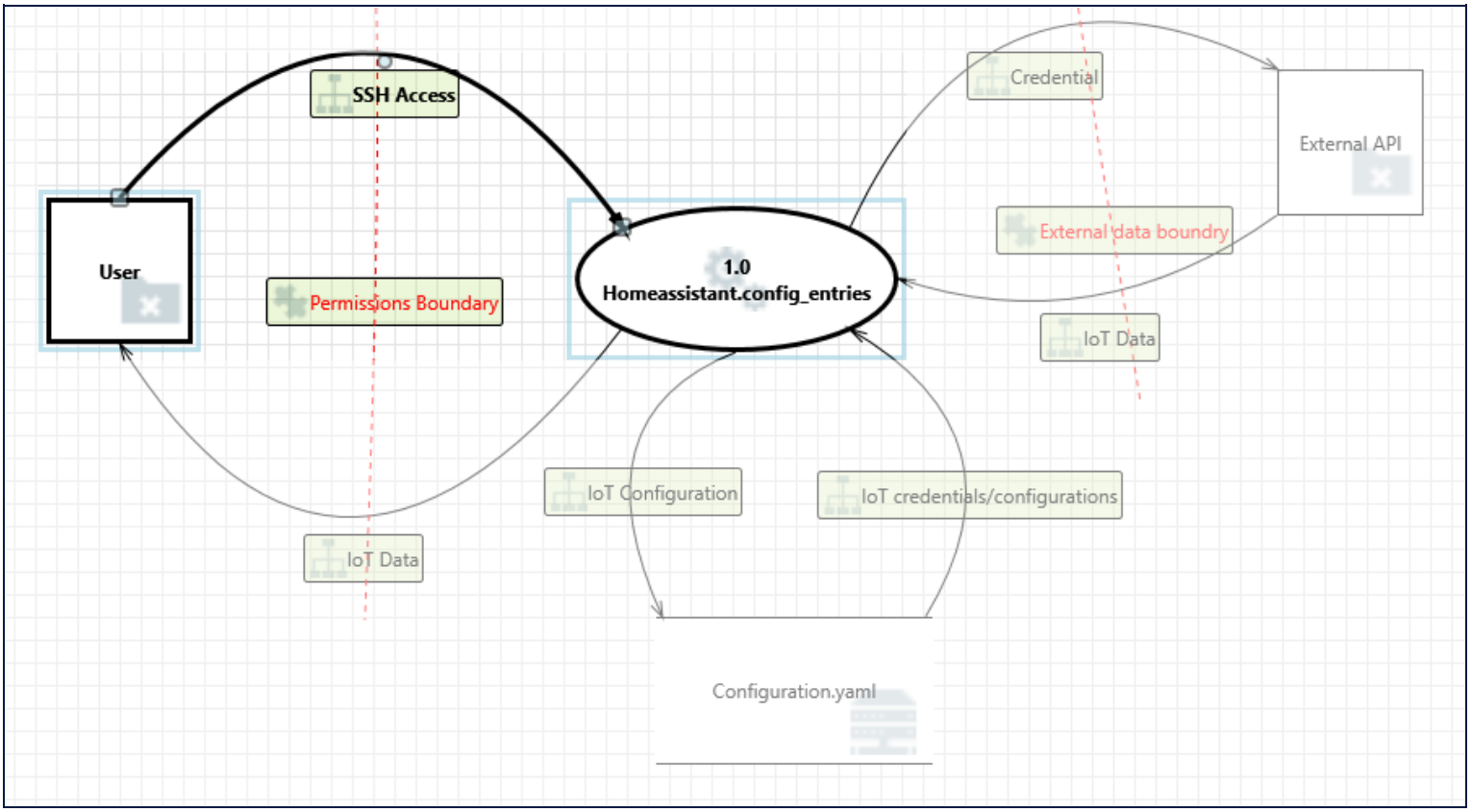
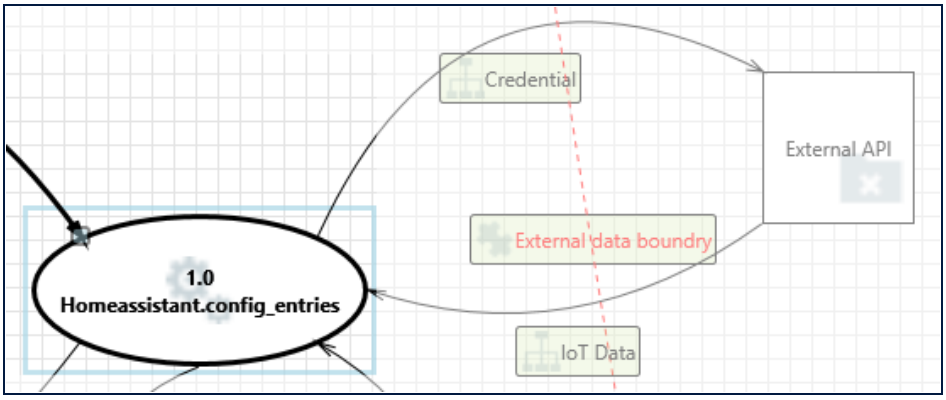


Diagram 1 Diagram Summary:

Not Started	0
Not Applicable	6
Needs Investigation	0
Mitigation Implemented	26
Total	32
Total Migrated	0

Interaction: Credential



1. Data Flow Credential Is Potentially Interrupted [State: Mitigation Implemented] [Priority: Low]

Category: Denial Of Service

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

Justification: It could be possible to interupt communication here, but highly unlikely. Also, It is possible some add-on APIs are on the local network.

2. External Entity External API Potentially Denies Receiving Data [State: Mitigation Implemented] [Priority: Low]

Category: Repudiation

Description: External API 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: The Home Assistant server provides logging of all request, responses, and actions made by the server.

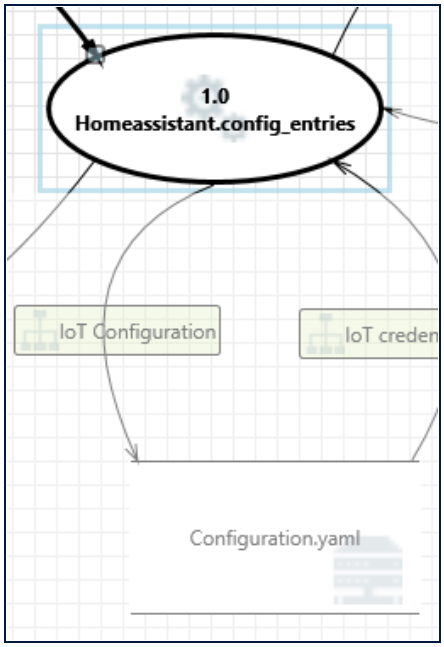
3. Spoofing of the External API External Destination Entity [State: Mitigation Implemented] [Priority: Medium]

Category: Spoofing

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

Justification: The Home Asstant uses the address in the configuration.yaml file to communicate with the External API, which is on a local network and access is protected by authentication.

Interaction: IoT Configuration



4. Potential Excessive Resource Consumption for 1.0 Homeassistant.config_entries or Configuration.yaml [State: Mitigation Implemented] [Priority: Medium]

Category: Denial Of Service

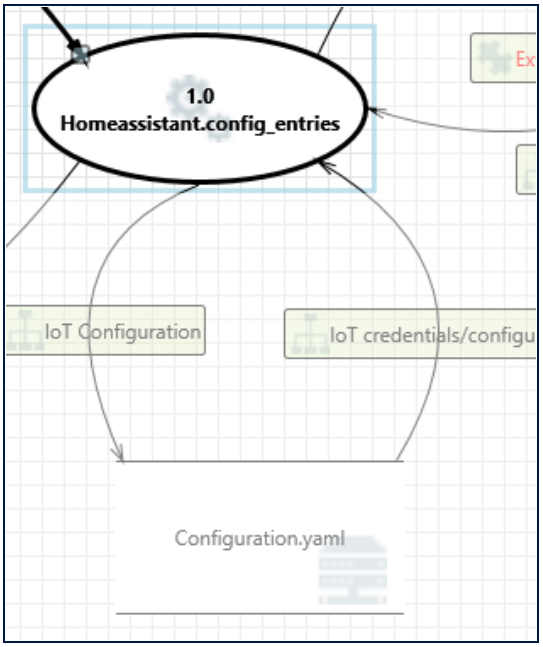
Description: Does 1.0 Homeassistant.config_entries or Configuration.yaml take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.

Justification: Home Assistant is currently configured to run multiple threads at once while allowing communication between the filesystem and the Home Assistant server without deadlock.

5. Spoofing of Destination Data Store Configuration.yaml [State: Mitigation Implemented] [Priority: Low]

Category: Spoofing
Description: Configuration.yaml may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of Configuration.yaml. Consider using a standard authentication mechanism to identify the destination data store.
Justification: The user must authenticate with username and password and must be on the local network to access the configuration.yaml file.

Interaction: IoT credentials/configurations



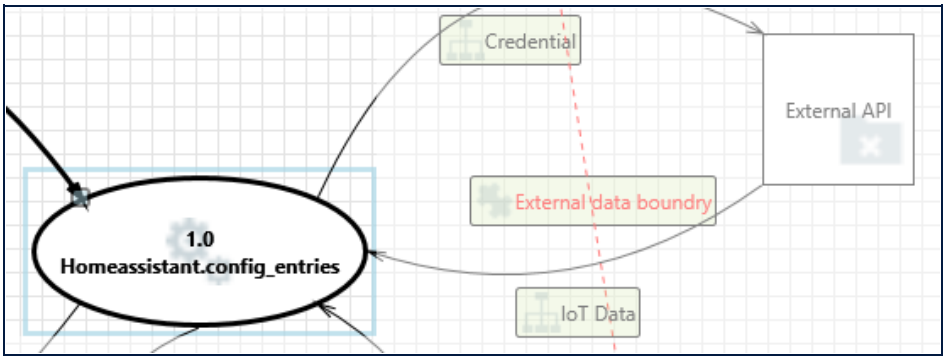
6. Weak Access Control for a Resource [State: Mitigation Implemented] [Priority: High]

Category: Information Disclosure
Description: Improper data protection of Configuration.yaml can allow an attacker to read information not intended for disclosure. Review authorization settings.
Justification: The Home Assistant provides various alternatives to save credentials in a safer way. It is recommended that users use a secrets.yaml file in addition to storing credentials in the keyring or using credstash, on an AWS server. However, this is not configured by default.

7. Spoofing of Source Data Store Configuration.yaml [State: Mitigation Implemented] [Priority: Low]

Category: Spoofing
Description: Configuration.yaml may be spoofed by an attacker and this may lead to incorrect data delivered to 1.0 Homeassistant.config_entries. Consider using a standard authentication mechanism to identify the source data store.
Justification: The Configuration.yaml file is on a local network and the user must authenticate to have access to this file.

Interaction: IoT Data



8. Cross Site Request Forgery [State: Mitigation Implemented] [Priority: Low]

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: The External API is authenticated using the username and password in the configuration.yaml file.

9. Elevation by Changing the Execution Flow in 1.0 Homeassistant.config_entries [State: Mitigation Implemented] [Priority: Low]

Category: Elevation Of Privilege
Description: An attacker may pass data into 1.0 Homeassistant.config_entries in order to change the flow of program execution within 1.0 Homeassistant.config_entries to the attacker's choosing.
Justification: The user is authenticated by username and password in configuration.yaml file before sending data.

10. 1.0 Homeassistant.config_entries May be Subject to Elevation of Privilege Using Remote Code Execution [State: Mitigation Implemented] [Priority: Low]

Category: Elevation Of Privilege
Description: External API may be able to remotely execute code for 1.0 Homeassistant.config_entries.
Justification: External API is authenticated with a username and password.

11. Elevation Using Impersonation [State: Mitigation Implemented] [Priority: Low]

Category: Elevation Of Privilege
Description: 1.0 Homeassistant.config_entries may be able to impersonate the context of External API in order to gain additional privilege.
Justification: The Home Assistant runs at the lowest privlege possible.

12. Data Flow IoT Data Is Potentially Interrupted [State: Not Applicable] [Priority: Low]

Category: Denial Of Service

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

Justification: There is no mitigation in place to prevent a denial of service of the external API, but if there is an inturuption there is extensive logging capabililty that the admin is able to review.

13. Potential Process Crash or Stop for 1.0 Homeassistant.config_entries [State: Mitigation Implemented] [Priority: Low]

Category: Denial Of Service

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

Justification: The Home Assistant server has snapshots that are able to be restored in case of a failure.

14. Data Flow Sniffing [State: Mitigation Implemented] [Priority: Medium]

Category: Information Disclosure

Description: Data flowing across IoT 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: The request is made in http, however a user can choose to set up https communication with Home Assistant.

15. Potential Data Repudiation by 1.0 Homeassistant.config_entries [State: Mitigation Implemented] [Priority: Low]

Category: Repudiation

Description: 1.0 Homeassistant.config_entries 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: The Home Assistant server provides logging of all request, responses, and actions made by the server.

16. Potential Lack of Input Validation for 1.0 Homeassistant.config_entries [State: Mitigation Implemented] [Priority: Low]

Category: Tampering

Description: Data flowing across IoT Data may be tampered with by an attacker. This may lead to a denial of service attack against 1.0 Homeassistant.config_entries or an elevation of privilege attack against 1.0 Homeassistant.config_entries or an information disclosure by 1.0 Homeassistant.config_entries. 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: The external API must be authericated with a username and password befrore sending the data, this account will use the lowest privlidges needed. Also, the configurations.yaml file will contain some information regarding what type of device it is and the input to expect.

17. Spoofing the External API External Entity [State: Mitigation Implemented] [Priority: Low]

Category: Spoofing

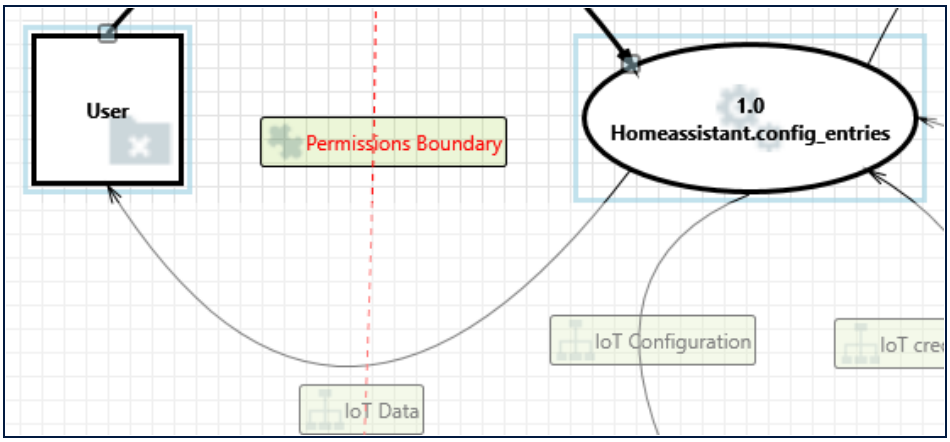
Description: External API may be spoofed by an attacker and this may lead to unauthorized access to 1.0 Homeassistant.config_entries. Consider using a standard authentication mechanism to identify the external entity.

Justification: The user is athenticated with a username and password.

18. Spoofing the 1.0 Homeassistant.config_entries Process [State: Mitigation Implemented] [Priority: Low]

Category: Spoofing
Description: 1.0 Homeassistant.config_entries may be spoofed by an attacker and this may lead to information disclosure by External API. Consider using a standard authentication mechanism to identify the destination process.
Justification: There is authentication via username and password to the external API.

Interaction: IoT Data



19. Data Flow IoT Data Is Potentially Interrupted [State: Not Applicable] [Priority: Low]

Category: Denial Of Service
Description: An external agent interrupts data flowing across a trust boundary in either direction.
Justification: The Home Assistant communication with the user usually happens on a local network.

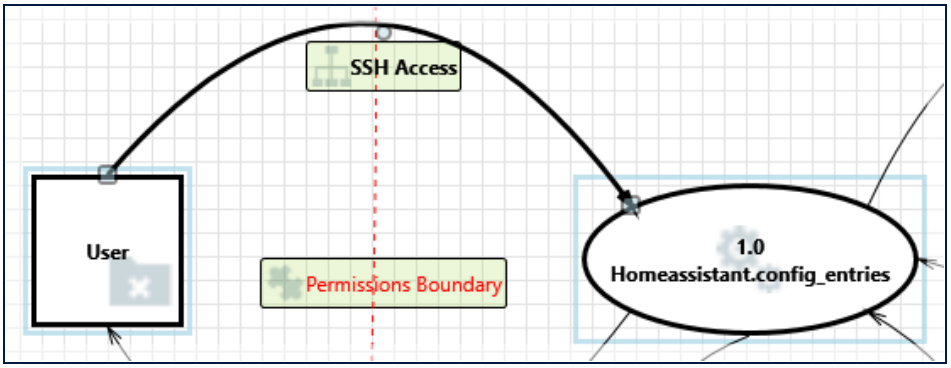
20. External Entity User Potentially Denies Receiving Data [State: Mitigation Implemented] [Priority: Low]

Category: Repudiation
Description: User 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: The Home Assistant server provides logging of all request, responses, and actions made by the server.

21. Spoofing of the User External Destination Entity [State: Mitigation Implemented] [Priority: Low]

Category: Spoofing
Description: User may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of User. Consider using a standard authentication mechanism to identify the external entity.
Justification: The connection between the user and the Home Assistant server is authenticated with username and password and is usuallt done on the local network.

Interaction: SSH Access



22. Potential Lack of Input Validation for 1.0 Homeassistant.config_entries [State: Not Applicable] [Priority: Low]

Category: Tampering

Description: Data flowing across IoT device add may be tampered with by an attacker. This may lead to a denial of service attack against 1.0 Homeassistant.config_entries or an elevation of privilege attack against 1.0 Homeassistant.config_entries or an information disclosure by 1.0 Homeassistant.config_entries. 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: Users must authenticate with a username and password or an SSH key when accessing via SSH. Otherwise, the user must be on the local network.

23. Spoofing the 1.0 Homeassistant.config_entries Process [State: Not Applicable] [Priority: Low]

Category: Spoofing

Description: 1.0 Homeassistant.config_entries may be spoofed by an attacker and this may lead to information disclosure by User. Consider using a standard authentication mechanism to identify the destination process.

Justification: Users must authenticate with a username and password or an SSH key when accessing via SSH. Otherwise, the user must be on the local network and authenticates the Home Assistant with a username and password.

24. Elevation Using Impersonation [State: Mitigation Implemented] [Priority: Medium]

Category: Elevation Of Privilege

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

Justification: Users must authenticate with a username and password or an SSH key when accessing via SSH. Otherwise, the user must be on the local network. Also, Home Assistant runs with least privlidges.

25. Spoofing the User External Entity [State: Mitigation Implemented] [Priority: Low]

Category: Spoofing

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

Justification: Users must authenticate with a username and password or an SSH key when accessing via SSH. Otherwise, the user must be on the local network.

26. Cross Site Request Forgery [State: Not Applicable] [Priority: Medium]

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: The communication of Home Assistant on usually on a local network but SSH can be enabled. By default the user is free to configure SSH as they wish. It is recommended users use SSH keys rather than passwords, but this is not configured by default.the agent would have to first gain access.

27. Elevation by Changing the Execution Flow in 1.0 Homeassistant.config_entries [State: Mitigation Implemented] [Priority: Medium]

Category: Elevation Of Privilege

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

Justification: The communication of Home Assistant on usually on a local network but SSH can be enabled. By default the user is free to configure SSH as they wish. It is recommended users use SSH keys rather than passwords, but this is not configured by default.

28. 1.0 Homeassistant.config_entries May be Subject to Elevation of Privilege Using Remote Code Execution [State: Mitigation Implemented] [Priority: Medium]

Category: Elevation Of Privilege

Description: User may be able to remotely execute code for 1.0 Homeassistant.config_entries.

Justification: The Home Assistant runs at the lowest privlidge possible to prevent this. Also, SSH must be authenticated with a username and password or an SSH key.

29. Data Flow IoT device add Is Potentially Interrupted [State: Mitigation Implemented] [Priority: Low]

Category: Denial Of Service

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

Justification: The communication of Home Assistant on usually on a local network but SSH can be enabled. It is recommended users use SSH keys rather than passwords, but this is not configured by default. If an agent gains access to the server via ssh it is possible to interupt communications.

30. Potential Process Crash or Stop for 1.0 Homeassistant.config_entries [State: Mitigation Implemented] [Priority: Low]

Category: Denial Of Service

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

Justification: The Home Assistant server has snapshops that are able to be restored in case of failure.

31. Data Flow Sniffing [State: Not Applicable] [Priority: Medium]

Category: Information Disclosure

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

Justification: The communication of the Home Assistant is usually done on a local network and the agent would need to access the network to tamper with requests. Enabling SSH increases the risk because it is recommended users use SSH keys rather than passwords, but this is not configured by default.

32. Potential Data Repudiation by 1.0 Homeassistant.config_entries [State: Mitigation Implemented] [Priority: Low]

Category: Repudiation

Description: 1.0 Homeassistant.config_entries 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: The Home Assistant server provides logging of all request, responses, and actions made by the server.