Threat Modeling Report

Created on 11/13/2020 3:39:08 PM

Threat Model Name:

Owner:

Reviewer:

Contributors:

Description:

Assumptions:

External Dependencies:

Threat Model Summary:

Not Started	0
Not Applicable	15
Needs Investigation	1
Mitigation Implemented	26
Total	42
Total Migrated	0

Diagram: Diagram 1

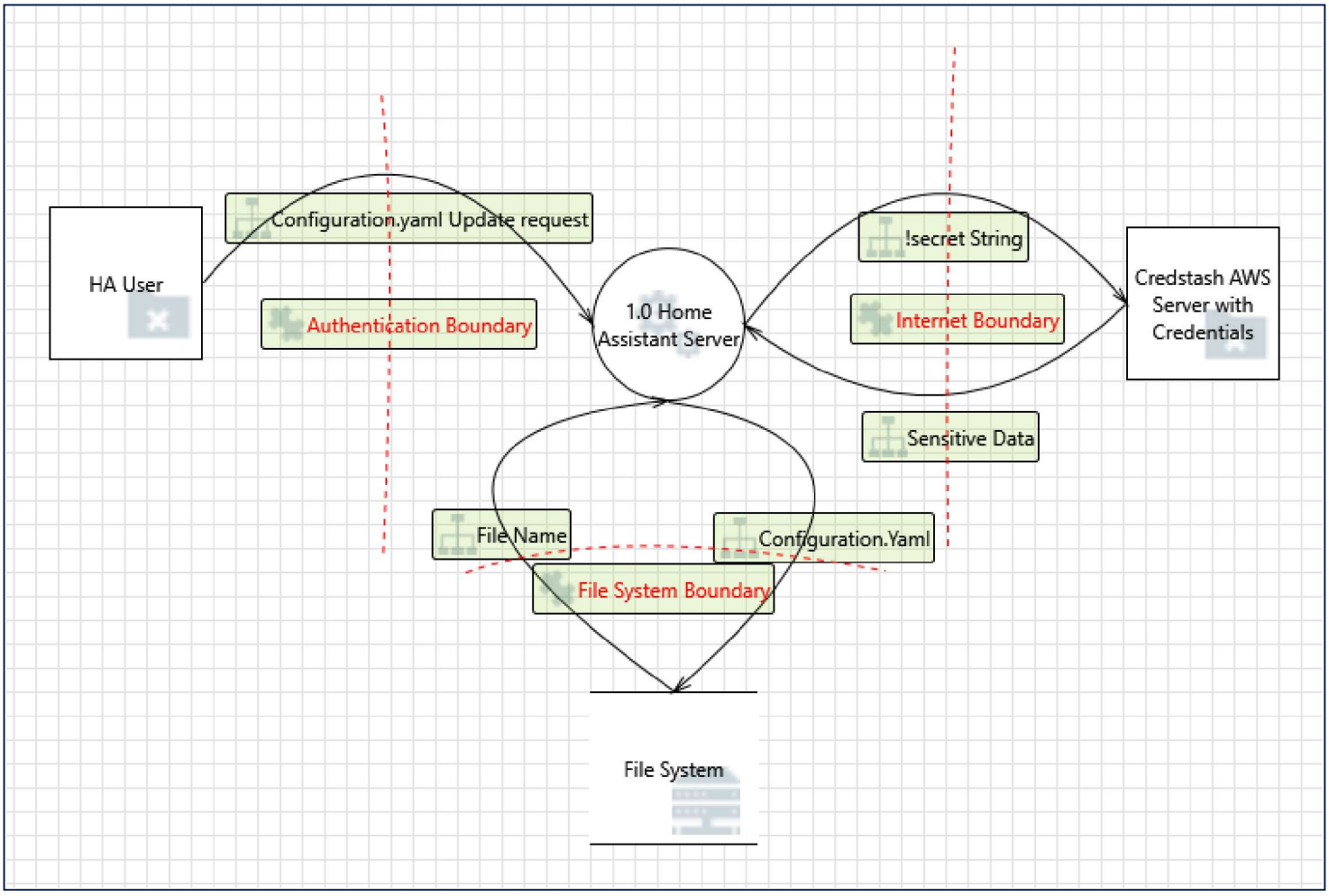
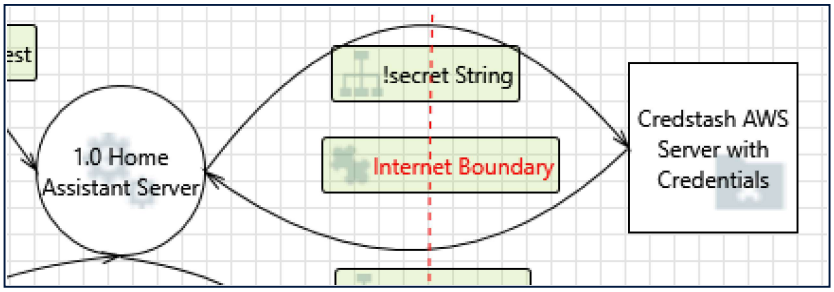


Diagram 1 Diagram Summary:

Not Started	0
Not Applicable	15
Needs Investigation	1
Mitigation Implemented	26
Total	42
Total Migrated	0

Interaction: !secret String



1. Spoofing of the Credstash AWS Server with Credentials External Destination Entity [State: Mitigation Implemented] [Priority: Medium]

Category: Spoofing

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

Justification: The AWS KMS key provides a method of encryption for the !secret String, an attacker will need to decrypt that key in order to compromise any data.

2. External Entity Credstash AWS Server with Credentials Potentially Denies Receiving Data [State: Mitigation Implemented] [Priority: Low]

Category: Repudiation

Description: Credstash AWS Server with Credentials 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 logs all communciation between Home Assistant and Credstash. The user can check the logs to see what happened.

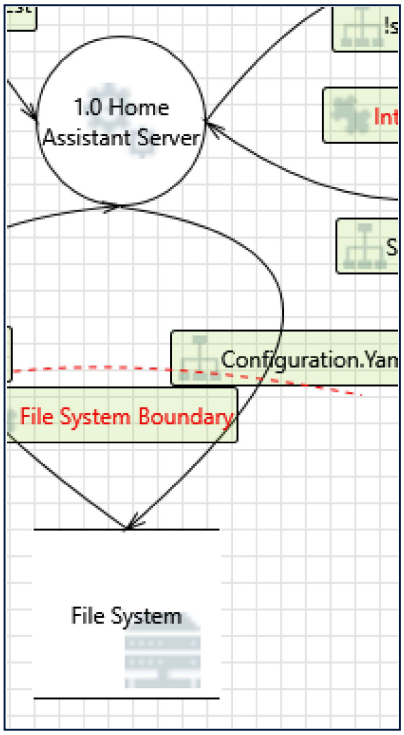
3. Data Flow !secret String Is Potentially Interrupted [State: Mitigation Implemented] [Priority: Medium]

Category: Denial Of Service

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

Justification: The data is encrypted, if it is interrupted the attacker cannot access any of the encrypted data without the key and Home Assistant will prompt the user to resubmit the request.

Interaction: Configuration.Yaml



4. Potential Excessive Resource Consumption for 1.0 Home Assistant Server or File System [State: Needs Investigation] [Priority: Medium]

Category: Denial Of Service

Description: Does 1.0 Home Assistant Server or File System 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: The Home Assistant Server is configured to run multiple processes at a single time allowing communication between the internal file system and the Home Assistant server to occur without deadlock.

5. Spoofing of Destination Data Store File System [State: Mitigation Implemented] [Priority: Low]

Category: Spoofing

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

Justification: The user will have to authenticate to the HA server in order for the server to retreive the Configuration.YAML file.

6. Spoofing the 1.0 Home Assistant Server Process [State: Mitigation Implemented] [Priority: Low]

Category: Spoofing

Description: 1.0 Home Assistant Server may be spoofed by an attacker and this may lead to unauthorized access to File System. Consider using a standard authentication mechanism to identify the source process.

Justification: Communication between the Home Assistant Server and the File System is done inside the OS. There is no way for an external agent to interrupt this data flow. Home Assistant also requires a username and password for root privileges.

7. The File System Data Store Could Be Corrupted [State: Not Applicable] [Priority: Low]

Category: Tampering

Description: Data flowing across Configuration.Yaml may be tampered with by an attacker. This may lead to corruption of File System. Ensure the integrity of the data flow to the data store.

Justification: Communication between the Home Assistant Server and the File System is done inside the OS. There is no way for an external agent to interrupt this data flow.

8. Data Store Denies File System Potentially Writing Data [State: Mitigation Implemented] [Priority: Medium]

Category: Repudiation

Description: File System 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: Home Assistant server provides detailed logging of all interactions that occur on the underlying

operating system.

9. Data Flow Sniffing [State: Mitigation Implemented] [Priority: Low]

Category: Information Disclosure

Description: Data flowing across Configuration.Yaml 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: Communication between the Home Assistant Server and the File System is done inside the OS. There is no way for an external agent to interrupt this data flow.

10. Data Flow Configuration.Yaml 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: Communication between the Home Assistant Server and the File System is done inside the OS. There is no way for an external agent to interrupt this data flow.

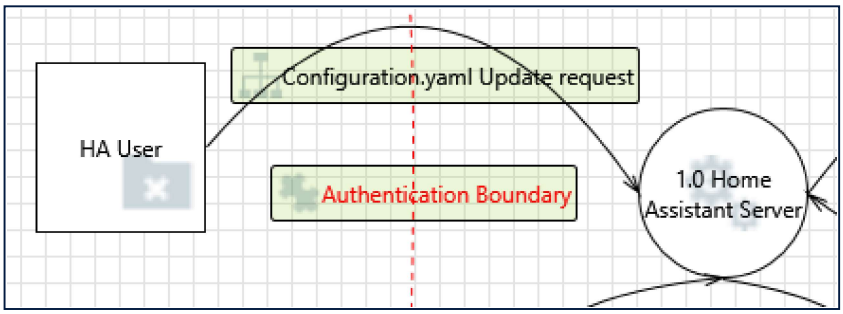
11. Data Store Inaccessible [State: Not Applicable] [Priority: Low]

Category: Denial Of Service

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

Justification: Communication between the Home Assistant Server and the File System is done inside the OS. There is no way for an external agent to prevent access.

Interaction: Configuration.yaml Update request



12. Spoofing the HA User External Entity [State: Not Applicable] [Priority: Low]

Category: Spoofing

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

Justification: The Home Assistant server is only applicable on the users local network.

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

Category: Elevation Of Privilege

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

Justification: The HA user authenticates to to the server with a username and password combination and runs in least privelge.

14. Cross Site Request Forgery [State: Not Applicable] [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: Communication between the Home Assistant Server and the user is done on the users local network. An attacker will have to gain access to the users local network.

15. Elevation by Changing the Execution Flow in 1.0 Home Assistant Server [State: Not Applicable] [Priority: Low]

Category: Elevation Of Privilege

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

Justification: Communication between the Home Assistant Server and the user is done on the local network which an attacker will not have access to.

16. 1.0 Home Assistant Server May be Subject to Elevation of Privilege Using Remote Code Execution [State: Mitigation Implemented] [Priority: Medium]

Category: Elevation Of Privilege

Description: HA User may be able to remotely execute code for 1.0 Home Assistant Server.

Justification: The HA user runs with the least amount of privileges possible.

17. Data Flow Configuration.yaml Update request 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: Communication between the Home Assistant Server and the File System is done on the local network which an attacker will not have access to.

18. Potential Process Crash or Stop for 1.0 Home Assistant Server [State: Mitigation Implemented] [Priority: Medium]

Category: Denial Of Service

Description: 1.0 Home Assistant Server crashes, halts, stops or runs slowly; in all cases violating an availability metric.

Justification: Home Assistant Server allows users to take snapshots of the device and they can restore the device to a previous snapshot in the event of a failure.

19. Data Flow Sniffing [State: Not Applicable] [Priority: Low]

Category: Information Disclosure

Description: Data flowing across Configuration.yaml Update 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: Communication between the HA User and the Home Assistant server is done on the local network. An attacker would need access to the users local network in order to tamper with the request.

20. Potential Data Repudiation by 1.0 Home Assistant Server [State: Mitigation Implemented] [Priority: Medium]

Category: Repudiation

Description: 1.0 Home Assistant Server claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.

Justification: The Home Assistant server provides detailed logging for all requests and actions performed on the server.

21. Potential Lack of Input Validation for 1.0 Home Assistant Server [State: Not Applicable] [Priority: Low]

Category: Tampering

Description: Data flowing across Configuration.yaml Update request may be tampered with by an attacker. This may lead to a denial of service attack against 1.0 Home Assistant Server or an elevation of privilege attack against 1.0 Home Assistant Server or an information disclosure by 1.0 Home Assistant Server. Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.

Justification: Communication between the HA User and the Home Assistant server is done on the local network. An attacker would need access to the users local network in order to tamper with the request.

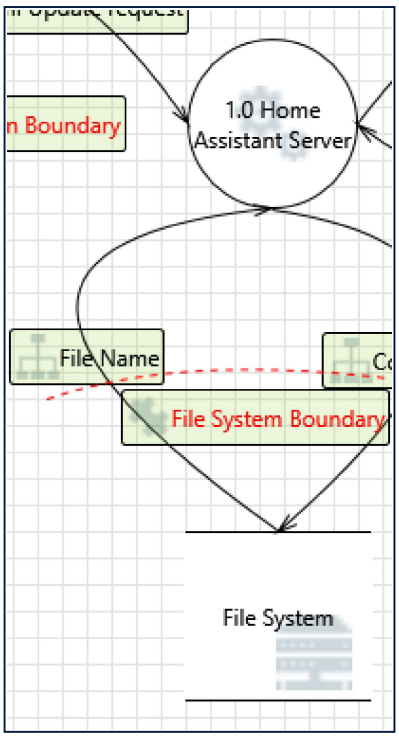
22. Spoofing the 1.0 Home Assistant Server Process [State: Not Applicable] [Priority: Low]

Category: Spoofing

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

Justification: The Home Assistant Server operates on the users local network, an attacker will need to gain access to the users local network in order to spoof Home Assistant.

Interaction: File Name



23. Weak Access Control for a Resource [State: Mitigation Implemented] [Priority: Low]

Category: Information Disclosure

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

Justification: Home Assistant also users to set permission using "chmod" for all files on the server.

24. Spoofing of Source Data Store File System [State: Not Applicable] [Priority: Low]

Category: Spoofing

Description: File System may be spoofed by an attacker and this may lead to incorrect data delivered to 1.0 Home Assistant Server. Consider using a standard authentication mechanism to identify the source data store.

Justification: The file systems is an interal part of the Home Assistant server which runs on the users local network. An attacker would have to get access to the users local network or have access to the device that

Home Assistant runs on

25. Spoofing the 1.0 Home Assistant Server Process [State: Not Applicable] [Priority: Low]

Category: Spoofing

Description: 1.0 Home Assistant Server may be spoofed by an attacker and this may lead to information disclosure by File System. Consider using a standard authentication mechanism to identify the destination process.

Justification: Home Assistant user operates on the users local network, an attacker would not have access to the local network unless they compromise the network which is out of scope.

26. Potential Data Repudiation by 1.0 Home Assistant Server [State: Mitigation Implemented] [Priority: Low]

Category: Repudiation

Description: 1.0 Home Assistant Server claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.

Justification: Home Assistant server provides detailed logging of all interactions that occur on the underlying operating system.

27. Potential Process Crash or Stop for 1.0 Home Assistant Server [State: Mitigation Implemented] [Priority: Low]

Category: Denial Of Service

Description: 1.0 Home Assistant Server crashes, halts, stops or runs slowly; in all cases violating an availability metric.

Justification: Home Assistant Server provides the user with the ability to take snapshots in which they can easily restore the server to the state it was before it crashed.

28. Data Flow File Name 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: Communication between the Home Assistant Server and the File System is done inside the OS. There is no way for an external agent to interrupt this data flow.

29. Data Store Inaccessible [State: Not Applicable] [Priority: Low]

Category: Denial Of Service

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

Justification: Communication between the Home Assistant Server and the File System is done inside the OS. There is no way for an external agent to interrupt this data flow.

30. 1.0 Home Assistant Server May be Subject to Elevation of Privilege Using Remote Code Execution [State: Mitigation Implemented] [Priority: Low]

Category: Elevation Of Privilege

Description: File System may be able to remotely execute code for 1.0 Home Assistant Server.

Justification: The file system is on the OS itself and does not have the necessary privileges to execute code on the host server.

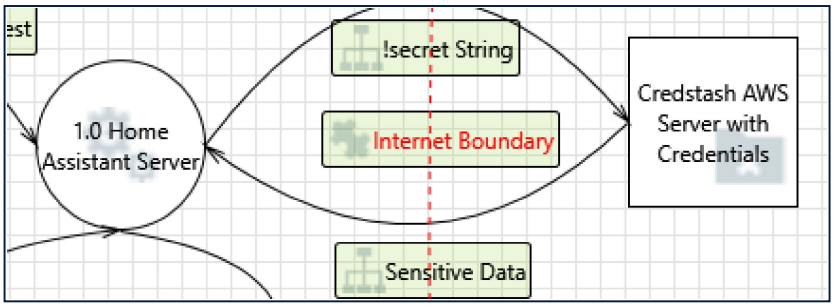
31. Elevation by Changing the Execution Flow in 1.0 Home Assistant Server [State: Not Applicable] [Priority: Low]

Category: Elevation Of Privilege

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

Justification: Communication between the Home Assistant Server and the File System is done inside the OS. There is no way for an external agent or attacker to interrupt this data flow.

Interaction: Sensitive Data



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

Category: Elevation Of Privilege

Description: 1.0 Home Assistant Server may be able to impersonate the context of Credstash AWS Server with Credentials in order to gain additional privilege.

Justification: The credentials used to authenticate to Credstash are different than what is needed to access "root" on Home Assistant.

33. Potential Lack of Input Validation for 1.0 Home Assistant Server [State: Mitigation Implemented] [Priority: High]

Category: Tampering

Description: Data flowing across Sensitive Data may be tampered with by an attacker. This may lead to a denial of service attack against 1.0 Home Assistant Server or an elevation of privilege attack against 1.0 Home Assistant Server or an information disclosure by 1.0 Home Assistant Server. Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the

way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.

Justification: Validation is done via a Amazons Key Management Service (KMS) master key ensuring that the data is not tampered and the input is verified for correctness since the master key resides on the AWS server and the wrapped encryption key to the KMS and the KMS decrypts it with the Master key.

34. Spoofing the 1.0 Home Assistant Server Process [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: 1.0 Home Assistant Server may be spoofed by an attacker and this may lead to information disclosure by Credstash AWS Server with Credentials. Consider using a standard authentication mechanism to identify the destination process.

Justification: Home Assistant Server authenticates itself to Credstash user a standard username and password combination.

35. Spoofing the Credstash AWS Server with Credentials External Entity [State: Mitigation Implemented] [Priority: Medium]

Category: Spoofing

Description: Credstash AWS Server with Credentials may be spoofed by an attacker and this may lead to unauthorized access to 1.0 Home Assistant Server. Consider using a standard authentication mechanism to identify the external entity.

Justification: Authentication to Credstash is done with a username and password combination.

36. Potential Data Repudiation by 1.0 Home Assistant Server [State: Mitigation Implemented] [Priority: Low]

Category: Repudiation

Description: 1.0 Home Assistant Server claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.

Justification: Home Assistant server provides detailed logging of all interactions that occur on the underlying operating system.

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

Category: Information Disclosure

Description: Data flowing across Sensitive 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 data that travels between credstash and Home Assistant is done via encrypted keys and over HTTPS.

38. Potential Process Crash or Stop for 1.0 Home Assistant Server [State: Mitigation Implemented] [Priority: Low]

Category: Denial Of Service

Description: 1.0 Home Assistant Server crashes, halts, stops or runs slowly; in all cases violating an availability metric.

Justification: Home Assistant allows the user to take snapshots so they can easily restore the OS to the state it was in before it crashed.

39. Data Flow Sensitive Data Is Potentially Interrupted [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

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

Justification: The data is encrytped with HTTPS and the AWS Key Management System so the data will not be compromised.

40. 1.0 Home Assistant Server May be Subject to Elevation of Privilege Using Remote Code Execution [State: Mitigation Implemented] [Priority: Medium]

Category: Elevation Of Privilege

Description: Credstash AWS Server with Credentials may be able to remotely execute code for 1.0 Home Assistant Server.

Justification: The AWS Key Management System ensures that no extra data is sent to and from Home Assistant Server and Credstash.

41. Elevation by Changing the Execution Flow in 1.0 Home Assistant Server [State: Mitigation Implemented] [Priority: Medium]

Category: Elevation Of Privilege

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

Justification: The AWS Key Management System ensures that no extra data is sent to and from Home Assistant Server and Credstash.

42. Cross Site Request Forgery [State: Mitigation Implemented] [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: The AWS Key Management System provides a way for the server to authenticate itself with Credstash and vice-versa.