Threat Modeling Report

Created on 11/13/2019 10:27:25 PM

Threat Model Name: Bitwarden

Owner: GotRoot

Reviewer:

Contributors: Ernewein, Robert; French, Scott; Minoungou, Toussida; Schmitz, Casey

Description:

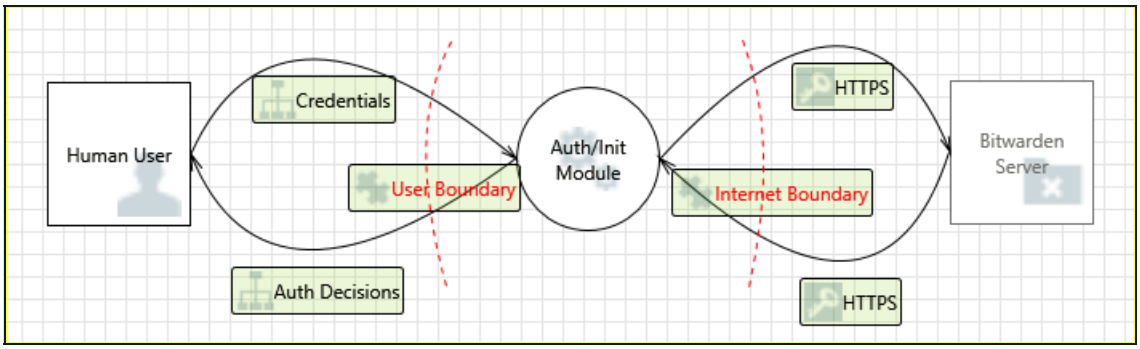
Assumptions: The Human User is using the Command-Line Bitwarden Client on a Windows machine.

External Dependencies:

Threat Model Summary:

Not Started	0
Not Applicable	35
Needs Investigation	8
Mitigation Implemented	9
Total	52
Total Migrated	0

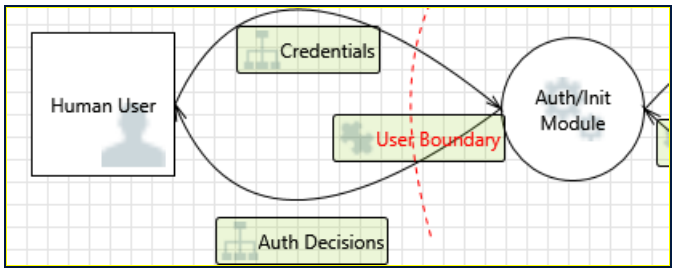
Diagram: L1 - Auth Module



L1 - Auth Module Diagram Summary:

Not Started	0
Not Applicable	14
Needs Investigation	5
Mitigation Implemented	3
Total	22
Total Migrated	0

Interaction: Auth Decisions



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

Category: Spoofing
Description: Human User may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of Human User. Consider using a standard authentication mechanism to identify the external entity.
Justification: Spoofing the Human Attacker's client would require physical access to open terminal.

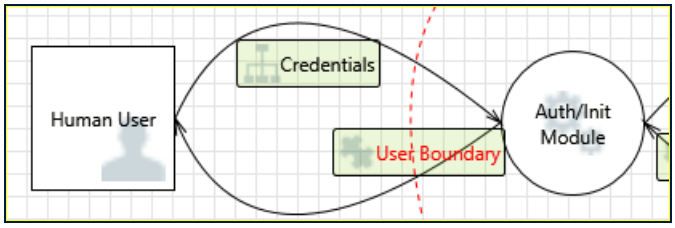
2. External Entity Human User Potentially Denies Receiving Data [State: Needs Investigation] [Priority: High]

Category: Repudiation
Description: Human 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: A one-way audit log can be referenced to assert whether transactions have occurred.

3. Data Flow Auth Decisions Is Potentially Interrupted [State: Needs Investigation] [Priority: High]

Category: Denial Of Service
Description: An external agent interrupts data flowing across a trust boundary in either direction.
Justification: <no mitigation provided>

Interaction: Credentials



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

Category: Elevation Of Privilege
Description: Auth Module may be able to impersonate the context of Human User in order to gain additional privilege.
Justification: Process executes with user permissions without privilege.

5. Spoofing the Auth Module Process [State: Not Applicable] [Priority: High]

Category: Spoofing
Description: Auth Module may be spoofed by an attacker and this may lead to information disclosure by Human User. Consider using a standard authentication mechanism to identify the destination process.
Justification: Can't prevent the user from attacking themselves.

6. Potential Lack of Input Validation for Auth Module [State: Not Applicable] [Priority: High]

Category: Tampering
Description: Data flowing across Credentials may be tampered with by an attacker. This may lead to a denial of service

attack against Auth Module or an elevation of privilege attack against Auth Module or an information disclosure by Auth Module. 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: An attacker would require physical access to the Human User's client.

7. Potential Data Repudiation by Auth Module [State: Needs Investigation] [Priority: High]

Category: Repudiation

Description: Auth Module 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: A one-way audit log can be referenced to assert whether transactions have occurred.

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

Category: Information Disclosure

Description: Data flowing across Credentials 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: An attacker would require physical access to the Human User's client.

9. Potential Process Crash or Stop for Auth Module [State: Not Applicable] [Priority: High]

Category: Denial Of Service

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

Justification: <no mitigation provided>

10. Data Flow Credentials 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: An attacker would require physical access to the Human User's client.

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

Category: Elevation Of Privilege

Description: Human User may be able to remotely execute code for Auth Module.

Justification: Process executes with user permissions without privilege.

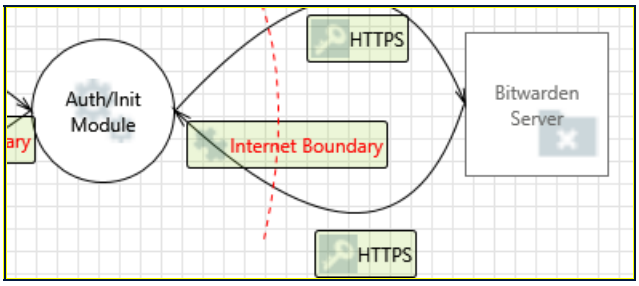
12. Elevation by Changing the Execution Flow in Auth Module [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

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

Justification: Process executes with user permissions without privilege.

Interaction: HTTPS



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

Category: Elevation Of Privilege
Description: Auth Module may be able to impersonate the context of Bitwarden Server in order to gain additional privilege.
Justification: Process executes with user permissions without privilege.

14. Potential Data Repudiation by Auth Module [State: Not Applicable] [Priority: High]

Category: Repudiation
Description: Auth Module 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: <no mitigation provided>

15. Potential Process Crash or Stop for Auth Module [State: Not Applicable] [Priority: High]

Category: Denial Of Service
Description: Auth Module crashes, halts, stops or runs slowly; in all cases violating an availability metric.
Justification: <no mitigation provided>

16. Data Flow HTTPS 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: <no mitigation provided>

17. Auth Module May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege
Description: Bitwarden Server may be able to remotely execute code for Auth/Init Module.
Justification: Process executes with user permissions without privilege.

18. Elevation by Changing the Execution Flow in Auth/Init Module [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege
Description: An attacker may pass data into Auth/Init Module in order to change the flow of program execution within Auth/Init Module to the attacker's choosing.
Justification: Process executes with user permissions without privilege.

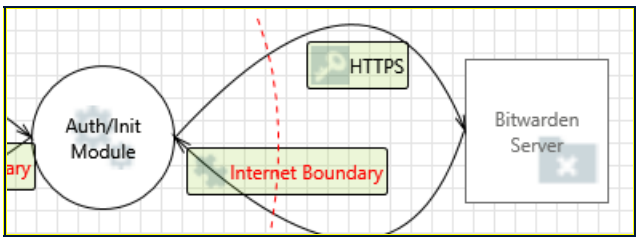
19. Cross Site Request Forgery [State: Not Applicable] [Priority: High]

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: Process executes with user permissions without privilege.

Interaction: HTTPS



20. Spoofing of the Bitwarden Server External Destination Entity [State: Needs Investigation] [Priority: High]

Category: Spoofing

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

Justification: Authentication of the Bitwarden Server can prevent this data from being sent to the attacker's target.

21. External Entity Bitwarden Server Potentially Denies Receiving Data [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Bitwarden Server 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: A one-way audit log can be referenced to assert whether transactions have occurred.

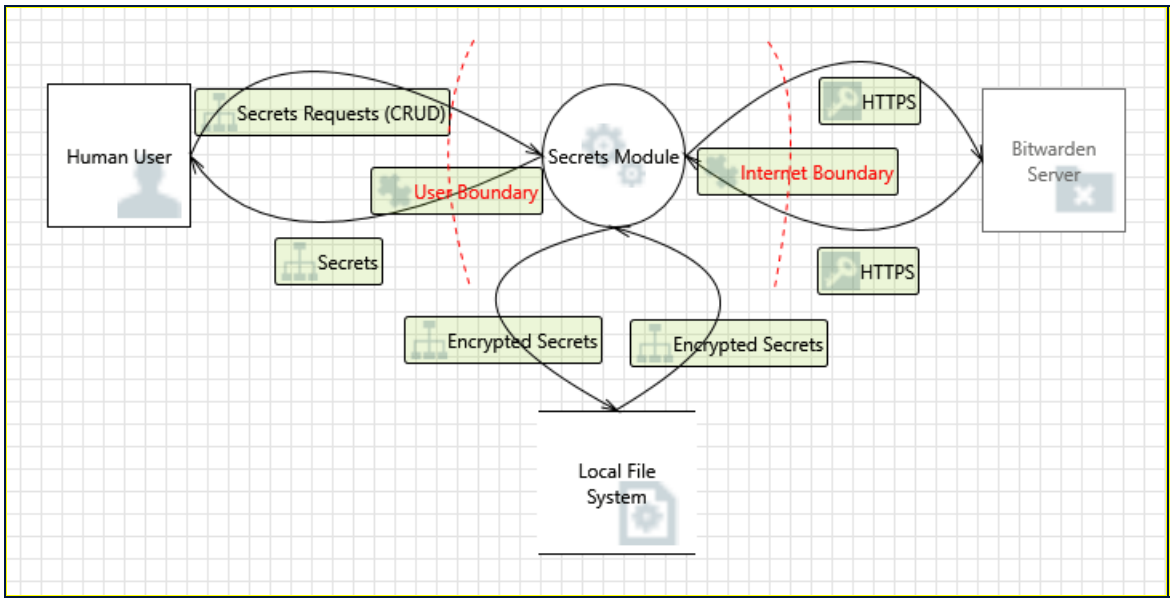
22. Data Flow HTTPS Is Potentially Interrupted [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

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

Justification: Monitors should be in place to ensure Auth Module does not consume excessive resources while Bitwarden Server is unavailable.

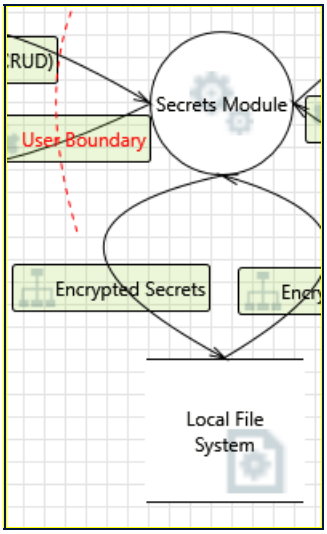
Diagram: L1 - Secrets Module



L1 - Secrets Module Diagram Summary:

Not Started	0
Not Applicable	21
Needs Investigation	3
Mitigation Implemented	6
Total	30
Total Migrated	0

Interaction: Encrypted Secrets



23. Weak Credential Storage [State: Mitigation Implemented] [Priority: High]

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: Master Password hash is encrypted using a one-way encryption algorithm.

24. Spoofing of Destination Data Store Local File System [State: Not Applicable] [Priority: High]

Category: Spoofing

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

Justification: Requires either local elevated privileges or access to the user account.

25. Authorization Bypass [State: Not Applicable] [Priority: High]

Category: Information Disclosure

Description: Can you access Local File System 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: Stored data is encrypted.

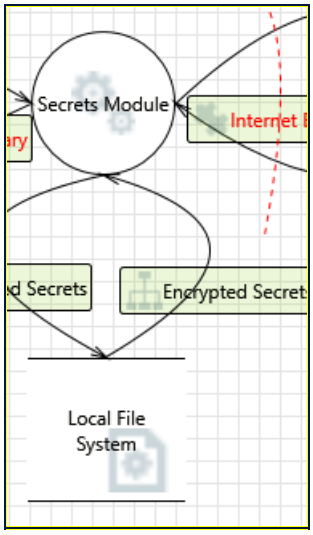
26. Potential Excessive Resource Consumption for Secrets Module or Local File System [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

Description: Does Secrets Module or Local 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: Secrets Module performs basic file I/O.

Interaction: Encrypted Secrets



27. Spoofing of Source Data Store Local File System [State: Not Applicable] [Priority: High]

Category: Spoofing

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

Justification: Requires either local elevated privileges or access to the user account.

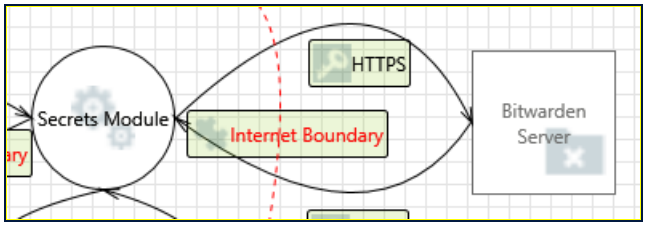
28. Weak Access Control for a Resource [State: Mitigation Implemented] [Priority: High]

Category: Information Disclosure

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

Justification: Stored data is encrypted.

Interaction: HTTPS



29. Spoofing of the Bitwarden Server External Destination Entity [State: Mitigation Implemented] [Priority: High]

Category: Spoofing
Description: Bitwarden Server may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of Bitwarden Server. Consider using a standard authentication mechanism to identify the external entity.
Justification: Authentication of the Bitwarden Server can prevent this encrypted data from being sent to the attacker's target.

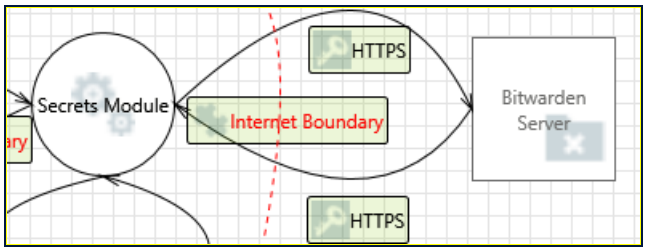
30. External Entity Bitwarden Server Potentially Denies Receiving Data [State: Mitigation Implemented] [Priority: High]

Category: Repudiation
Description: Bitwarden Server 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: A one-way audit log can be referenced to assert whether transactions have occurred.

31. Data Flow HTTPS Is Potentially Interrupted [State: Needs Investigation] [Priority: High]

Category: Denial Of Service
Description: An external agent interrupts data flowing across a trust boundary in either direction.
Justification: Monitors should be in place to ensure Secrets Module does not consume excessive resources while Bitwarden Server is unavailable.

Interaction: HTTPS



32. Potential Data Repudiation by Secrets Module [State: Mitigation Implemented] [Priority: High]

Category: Repudiation
Description: Secrets Module 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: A one-way audit log can be referenced to assert whether transactions have occurred.

33. Potential Process Crash or Stop for Secrets Module [State: Not Applicable] [Priority: High]

Category: Denial Of Service
Description: Secrets Module crashes, halts, stops or runs slowly; in all cases violating an availability metric.
Justification: <no mitigation provided>

34. Data Flow HTTPS 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: <no mitigation provided>

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

Category: Elevation Of Privilege
Description: Secrets Module may be able to impersonate the context of Bitwarden Server in order to gain additional privilege.
Justification: Process executes with user permissions without privilege.

36. Secrets Module May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege
Description: Bitwarden Server may be able to remotely execute code for Secrets Module.
Justification: Process executes with user permissions without privilege.

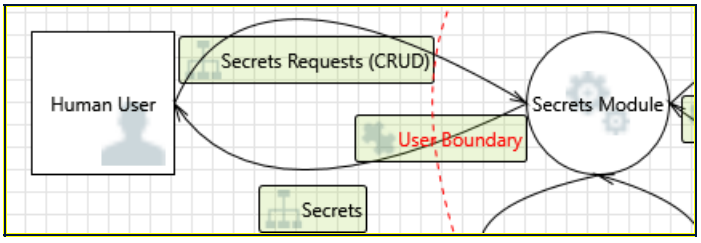
37. Elevation by Changing the Execution Flow in Secrets Module [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege
Description: An attacker may pass data into Secrets Module in order to change the flow of program execution within Secrets Module to the attacker's choosing.
Justification: Process executes with user permissions without privilege.

38. 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: Process executes with user permissions without privilege.

Interaction: Secrets



39. Spoofing of the Human User External Destination Entity [State: Not Applicable] [Priority: High]

Category: Spoofing

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

Justification: Spoofing the Human Attacker's client would require physical access to open terminal.

40. External Entity Human User Potentially Denies Receiving Data [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Human 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: A one-way audit log can be referenced to assert whether transactions have occurred.

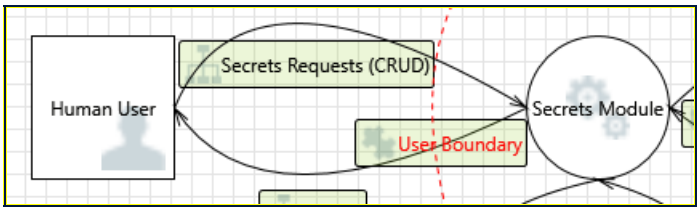
41. Data Flow Secrets Is Potentially Interrupted [State: Not Applicable] [Priority: High]

Category: Denial Of Service

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

Justification: <no mitigation provided>

Interaction: Secrets Requests (CRUD)



42. Spoofing the Secrets Module Process [State: Not Applicable] [Priority: High]

Category: Spoofing

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

Justification: Can't prevent the user from attacking themselves.

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

Category: Tampering

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

Justification: An attacker would require physical access to the Human User's client.

44. Potential Lack of Input Validation for Secrets Module [State: Not Applicable] [Priority: High]

Category: Tampering

Description: Data flowing across Secrets Requests (CRUD) may be tampered with by an attacker. This may lead to a denial of service attack against Secrets Module or an elevation of privilege attack against Secrets Module or an information disclosure by Secrets Module. 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: An attacker would require physical access to the Human User's client.

45. Potential Data Repudiation by Secrets Module [State: Not Applicable] [Priority: High]

Category: Repudiation

Description: Secrets Module 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: A one-way audit log can be referenced to assert whether transactions have occurred.

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

Category: Information Disclosure

Description: Data flowing across Secrets Requests (CRUD) 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: A one-way audit log can be referenced to assert whether transactions have occurred.

47. Potential Process Crash or Stop for Secrets Module [State: Not Applicable] [Priority: High]

Category: Denial Of Service

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

Justification: <no mitigation provided>

48. Data Flow Secrets Requests (CRUD) Is Potentially Interrupted [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

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

Justification: Monitors should be in place to ensure Secrets Module does not consume excessive resources while Bitwarden Server is unavailable.

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

Category: Elevation Of Privilege

Description: Secrets Module may be able to impersonate the context of Human User in order to gain additional privilege.

Justification: Process executes with user permissions without privilege.

50. Secrets Module May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

Description: Human User may be able to remotely execute code for Secrets Module.

Justification: Process executes with user permissions without privilege.

51. Elevation by Changing the Execution Flow in Secrets Module [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

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

Justification: Process executes with user permissions without privilege.

52. 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: Process executes with user permissions without privilege.