Threat Modeling Report

Created on 11/13/2019 11:13:36 PM

Threat Model Name:

Owner:

Reviewer:

Contributors:

Description:

Assumptions:

External Dependencies:

Threat Model Summary:

Not Started	0
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	18
Total	18
Total Migrated	0

Diagram: Diagram 1

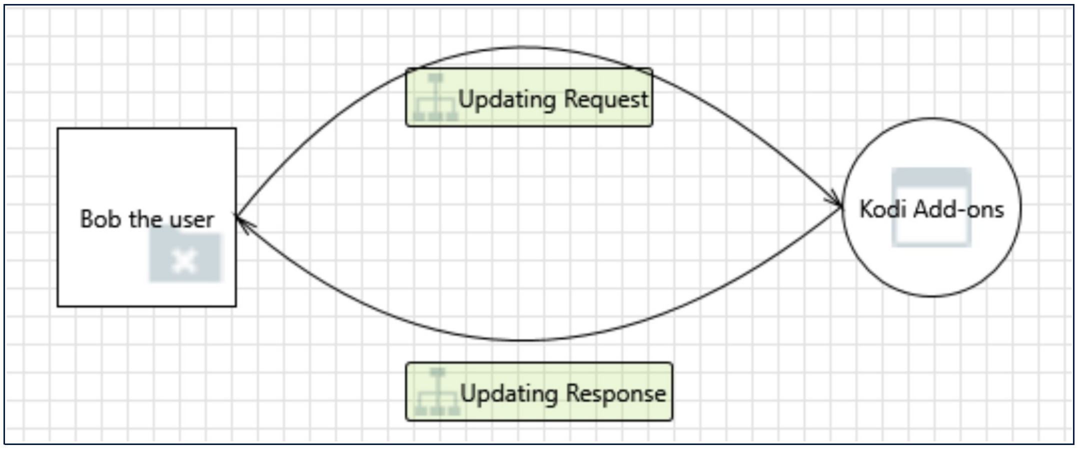
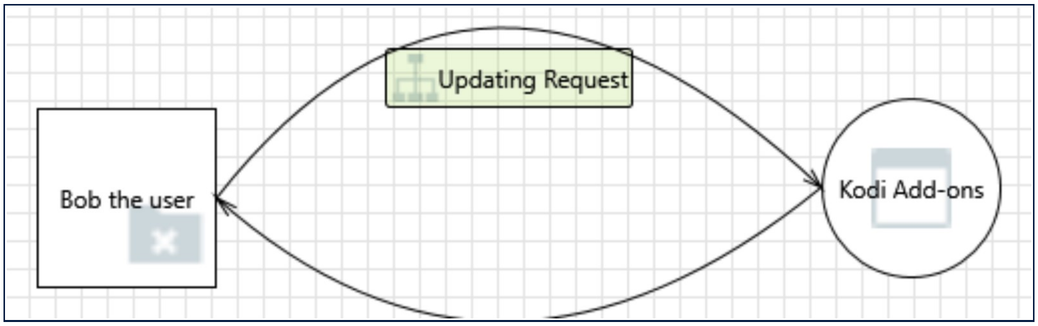


Diagram 1 Diagram Summary:

Not Started	0
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	2
Total	2
Total Migrated	0

Interaction: Updating Request



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

Category:	Spoofing
Description:	Bob the user may be spoofed by an attacker and this may lead to unauthorized access to Kodi Add-ons. Consider using a standard authentication mechanism to identify the external entity.
Justification:	Bob the user should set a PIN that will be his authentication.

2. Elevation Using Impersonation [State: Mitigation Implemented] [Priority: High]

Category:	Elevation Of Privilege
Description:	Kodi Add-ons may be able to impersonate the context of Bob the user in order to gain additional privilege.
Justification:	Kodi users can use the Master Lock to mitigate the elevation of privileges.

Diagram: Diagram 2

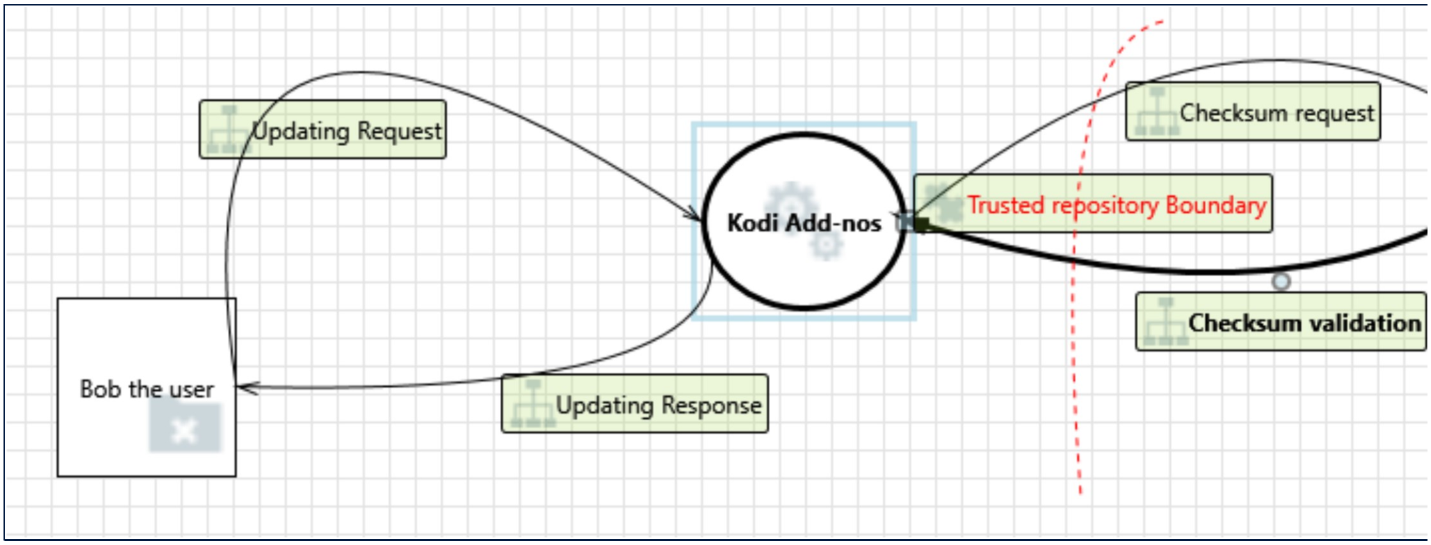
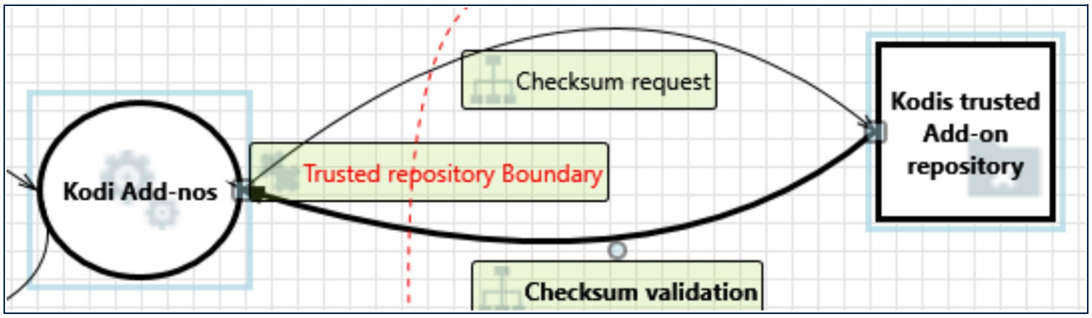


Diagram 2 Diagram Summary:

Not Started	0
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	16
Total	16
Total Migrated	0

Interaction: Checksum request



3. Spoofing of the Kodis trusted Add-on repositiory External Destination Entity [State: Mitigation Implemented] [Priority: High]

Category:	Spoofing
Description:	Kodis trusted Add-on repository may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of Kodis trusted Add-on repository. Consider using a standard authentication mechanism to identify the external entity.
Justification:	Kodi has a list of trusted repositories that have been set up with mitigating procedures in place.

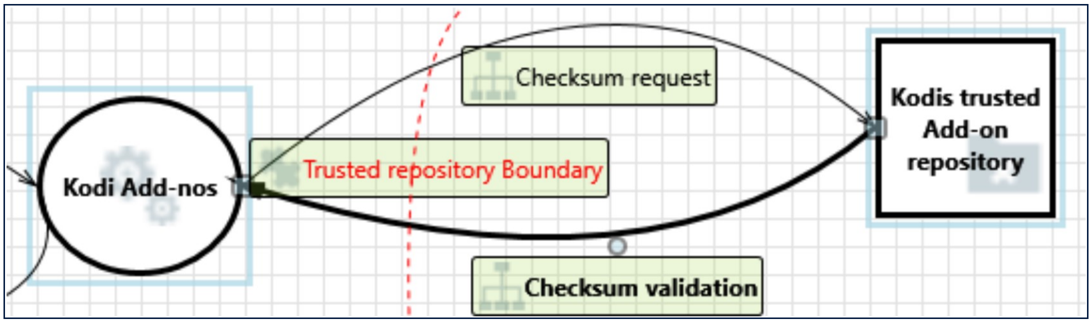
4. External Entity Kodis trusted Add-on repositiory Potentially Denies Receiving Data [State: Mitigation Implemented] [Priority: High]

Category:	Repudiation
Description:	Kodis trusted Add-on repository 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:	There is not an expected responce from the repository. If the checksum is the same then no data will be transmitted, if the checksum is different then a file is pulled.

5. Data Flow Checksum request 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:	Kodi has a list of trusted repositories that have been set up with mitigating procedures in place. The checksum should be within set tolerance and would be known by Kodi.

Interaction: Checksum validation



6. Elevation Using Impersonation [State: Mitigation Implemented] [Priority: High]

Category:	Elevation Of Privilege
Description:	Kodi Add-nos may be able to impersonate the context of Kodis trusted Add-on repositiory in order to gain additional privilege.
Justification:	Kodi users can use the Master Lock to mitigate the elevation of privileges.

7. Spoofing the Kodis tursted Add-on repositiory External Entity [State: Mitigation Implemented] [Priority: High]

Category:	Spoofing

Description: Kodis trusted Add-on repositiory may be spoofed by an attacker and this may lead to unauthorized access to Kodi Add-nos. Consider using a standard authentication mechanism to identify the external entity.

Justification: Kodi has a list of trusted repositories that have been set up with mitigating procedures in place.

8. Spoofing the Kodi Add-nos Process [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Kodi Add-nos may be spoofed by an attacker and this may lead to information disclosure by Kodis trusted Add-on repository. Consider using a standard authentication mechanism to identify the destination process.

Justification: Kodi uses either HTTPS or SSL with all of the trusted repositorise with none of them sending data outside of the checksum request.

9. Potential Lack of Input Validation for Kodi Add-nos [State: Mitigation Implemented] [Priority: High]

Category: Tampering

Description: Data flowing across Checksum validation may be tampered with by an attacker. This may lead to a denial of service attack against Kodi Add-nos or an elevation of privilege attack against Kodi Add-nos or an information disclosure by Kodi Add-nos. 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: Kodi uses either HTTPS or SSL with all of the trusted repositorise with none of them sending data outside of the checksum request.

10. Potential Data Repudiation by Kodi Add-nos [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Kodi Add-nos 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: There is not an expected responce from the repository. If the checksum is the same then no data will be transmitted, if the checksum is different then a file is pulled.

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

Category: Information Disclosure

Description: Data flowing across Checksum validation 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 only data being sent across this data flow is only a file that has been changed for the update. No other information will be sent with it. No, user info is used in this data flow.

12. Potential Process Crash or Stop for Kodi Add-nos [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

Description: Kodi Add-nos crashes, halts, stops or runs slowly; in all cases violating an availability metric.

Justification: If the data flow is interrupted that will only make the system check back at a later time.

13. Data Flow Checksum validation 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: If the data flow is interrupted that will only make the system check back at a later time.

14. Kodi Add-nos May be Subject to Elevation of Privilege Using Remote Code Execution [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: Kodis trusted Add-on repository may be able to remotely execute code for Kodi Add-nos.

Justification: Kodi users can use the Master Lock to mitigate the elevation of privileges.

15. Elevation by Changing the Execution Flow in Kodi Add-nos [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: An attacker may pass data into Kodi Add-nos in order to change the flow of program execution within Kodi Add-nos to the attacker's choosing.

Justification: Kodi users can use the Master Lock to mitigate the elevation of privileges.

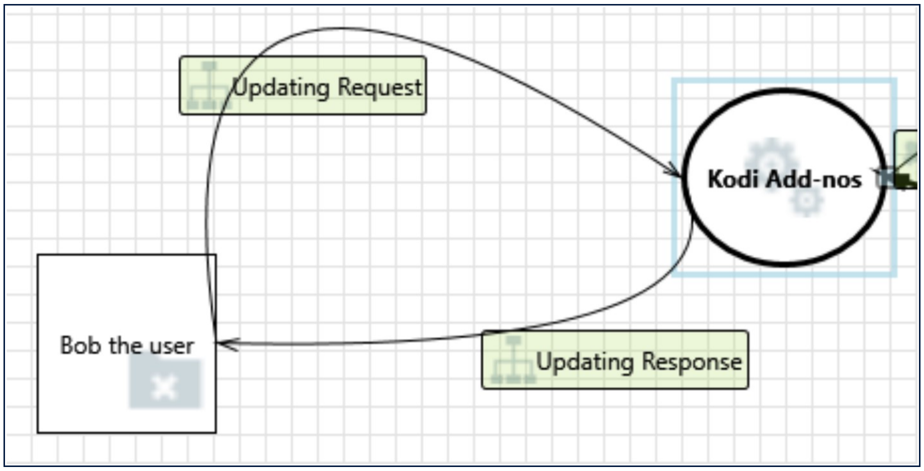
16. Cross Site Request Forgery [State: Mitigation Implemented] [Priority: High]

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: Kodi uses either HTTPS or SSL with all of the trusted repositorise with none of them sending data outside of the checksum request.

Interaction: Updating Request



17. Elevation Using Impersonation [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: Kodi Add-nos may be able to impersonate the context of Bob the user in order to gain additional privilege.

Justification: Kodi users can use the Master Lock to mitigate the elevation of privileges.

18. Spoofing the Bob the user External Entity [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Bob the user may be spoofed by an attacker and this may lead to unauthorized access to Kodi Add-nos. Consider using a standard authentication mechanism to identify the external entity.

Justification: Kodi has a PIN used to validate users.