Threat Modeling Report

Created on 11/13/2019 10:41:28 PM

Threat Model Name:

Owner:

Reviewer:

Contributors:

Description:

Assumptions:

External Dependencies:

Threat Model Summary:

Not Started	0
Not Applicable	4
Needs Investigation	0
Mitigation Implemented	18
Total	22
Total Migrated	0

Diagram: Diagram 1

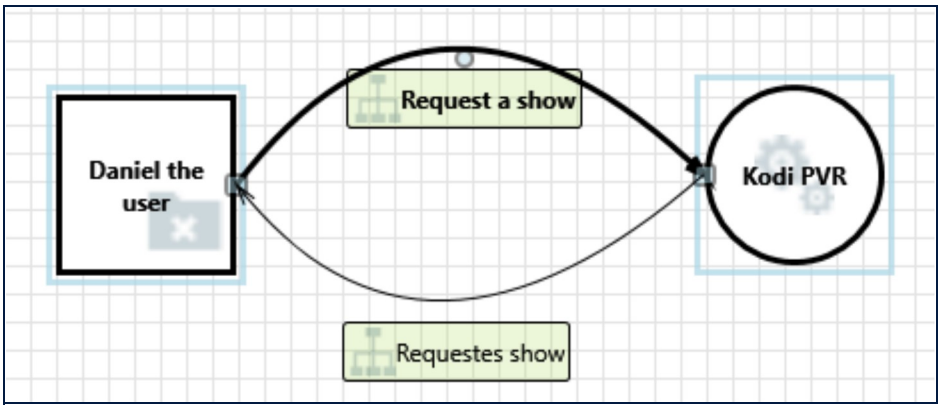
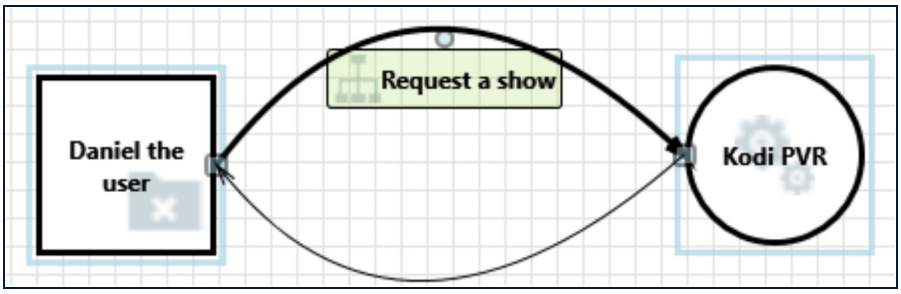


Diagram 1 Diagram Summary:

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

Interaction: Request a show



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

Category:	Spoofing
Description:	Daniel the user may be spoofed by an attacker and this may lead to unauthorized access to Kodi PVR. Consider using a standard authentication mechanism to identify the external entity.
Justification:	Kodi has a PIN used to validate users.

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

Category:	Elevation Of Privilege
Description:	Kodi PVR may be able to impersonate the context of Daniel 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

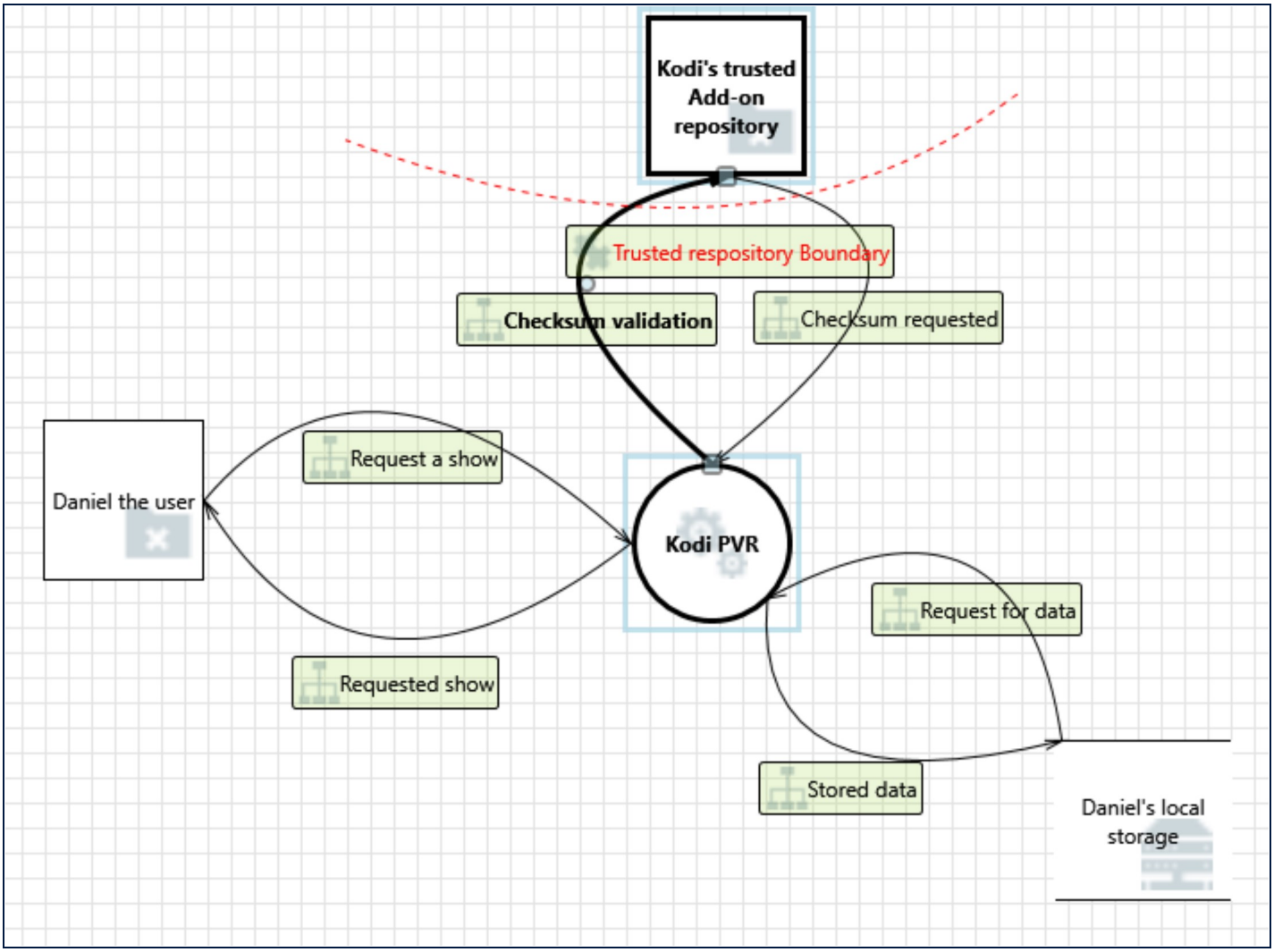
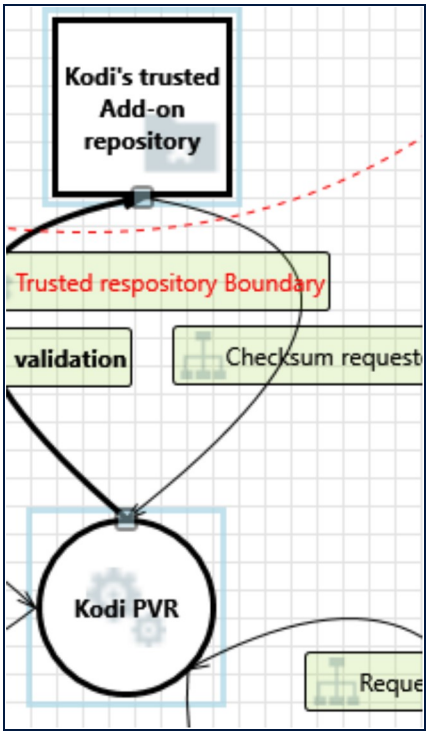


Diagram 2 Diagram Summary:

Not Started	0
Not Applicable	4
Needs Investigation	0
Mitigation Implemented	16
Total	20
Total Migrated	0

Interaction: Checksum requested



3. Spoofing the Kodi's trusted Add-on repository External Entity [State: Mitigation Implemented] [Priority: High]

Category:	Spoofing
Description:	Kodi's trusted Add-on repository may be spoofed by an attacker and this may lead to unauthorized access to Kodi PVR. 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. Elevation Using Impersonation [State: Mitigation Implemented] [Priority: High]

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

5. Spoofing the Kodi PVR Process [State: Mitigation Implemented] [Priority: High]

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

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

6. Potential Lack of Input Validation for Kodi PVR [State: Mitigation Implemented] [Priority: High]

Category: Tampering

Description: Data flowing across Checksum requested may be tampered with by an attacker. This may lead to a denial of service attack against Kodi PVR or an elevation of privilege attack against Kodi PVR or an information disclosure by Kodi PVR. 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 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.

7. Potential Data Repudiation by Kodi PVR [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Kodi PVR 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: 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.

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

Category: Information Disclosure

Description: Data flowing across Checksum requested 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: 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.

9. Potential Process Crash or Stop for Kodi PVR [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

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

Justification: Kodi uses Logs and communicates to its users when a service crashes, halts, stops, or runs slowly.

10. Data Flow Checksum requested 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.

11. Kodi PVR May be Subject to Elevation of Privilege Using Remote Code Execution [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: Kodi's trusted Add-on repository may be able to remotely execute code for Kodi PVR.

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

12. Elevation by Changing the Execution Flow in Kodi PVR [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

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

Justification: This process this handled through Kodi with set parameters and file list.

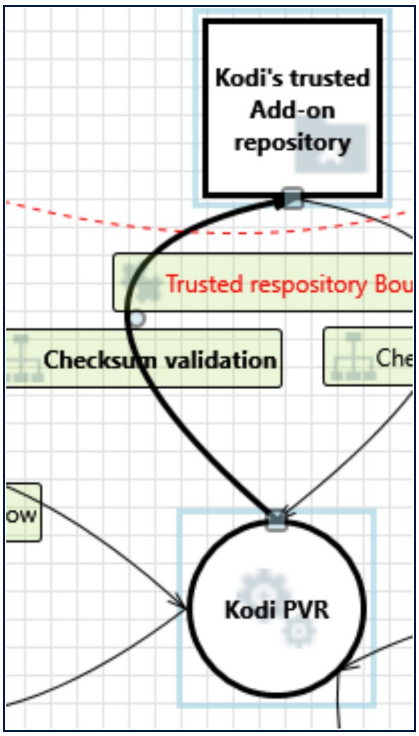
13. Cross Site Request Forgery [State: Mitigation Implemented] [Priority: High]

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: Checksum validation



14. Spoofing of the Kodi's trusted Add-on repository External Destination Entity [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Kodi's 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 Kodi's trusted Add-on repository. Consider using a standard authentication mechanism to identify the external entity.

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

15. External Entity Kodi's trusted Add-on repository Potentially Denies Receiving Data [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Kodi's 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.

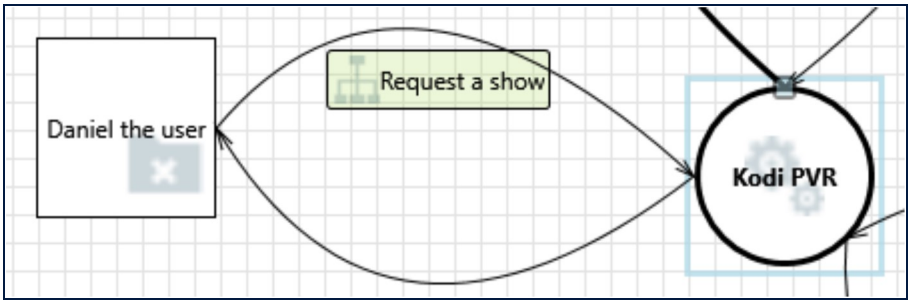
16. Data Flow Checksum validation Is Potentially Interrupted [State: Mitigation Implemented] [Priority: High]

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.

Interaction: Request a show



17. Spoofing the Daniel the user External Entity [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

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

Justification: Kodi has a PIN used to validate users.

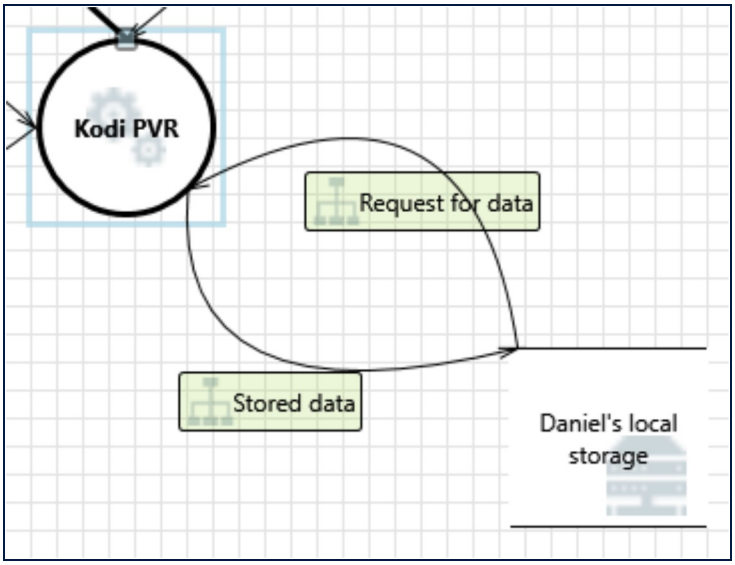
18. Elevation Using Impersonation [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: Kodi PVR may be able to impersonate the context of Daniel the user in order to gain additional privilege.

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

Interaction: Request for data



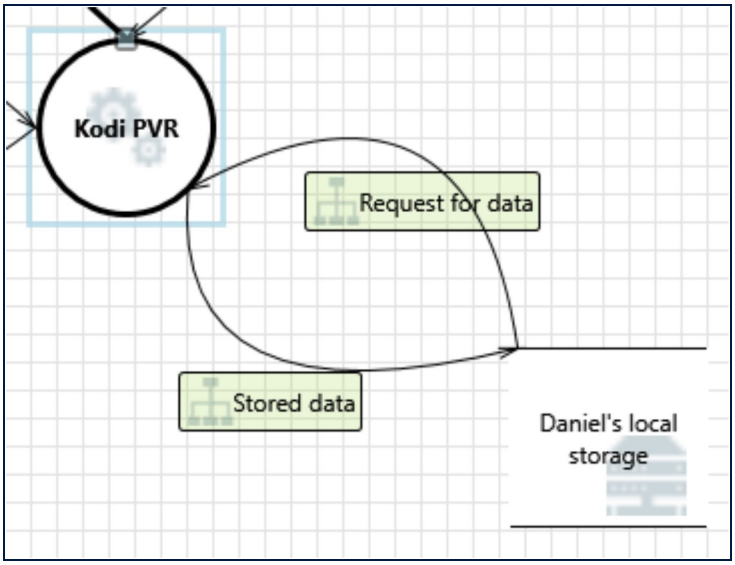
19. Weak Access Control for a Resource [State: Not Applicable] [Priority: High]

Category:	Information Disclosure
Description:	Improper data protection of Daniel's local storage can allow an attacker to read information not intended for disclosure. Review authorization settings.
Justification:	Daniel's local storage is outside the scope of this review. Daniel would need to implement security measures to ensure data integrity.

20. Spoofing of Source Data Store Generic Data Store [State: Not Applicable] [Priority: High]

Category:	Spoofing
Description:	Daniel's local storage may be spoofed by an attacker and this may lead to incorrect data delivered to Kodi PVR. Consider using a standard authentication mechanism to identify the source data store.
Justification:	Daniel's local storage is outside the scope of this review. Daniel would need to implement security measures to ensure data integrity.

Interaction: Stored data



21. Spoofing of Destination Data Store Generic Data Store [State: Not Applicable] [Priority: High]

Category:	Spoofing
Description:	Daniel's local storage may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of Daniel's local storage. Consider using a standard authentication mechanism to identify the destination data store.
Justification:	Daniel's local storage is outside the scope of this review. Daniel would need to implement security measures to ensure data integrity.

22. Potential Excessive Resource Consumption for Kodi PVR or Generic Data Store [State: Not Applicable] [Priority: High]

Category:	Denial Of Service
Description:	Does Kodi PVR or Daniel's local storage 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:	Daniel's local storage is outside the scope of this review. Daniel would need to implement security measures to ensure data integrity and resoutce consumption attacks.