Threat Modeling Report

Created on 11/8/2017 9:06:57 PM

Threat Model Name:

Owner:

Reviewer:

Contributors:

Description:

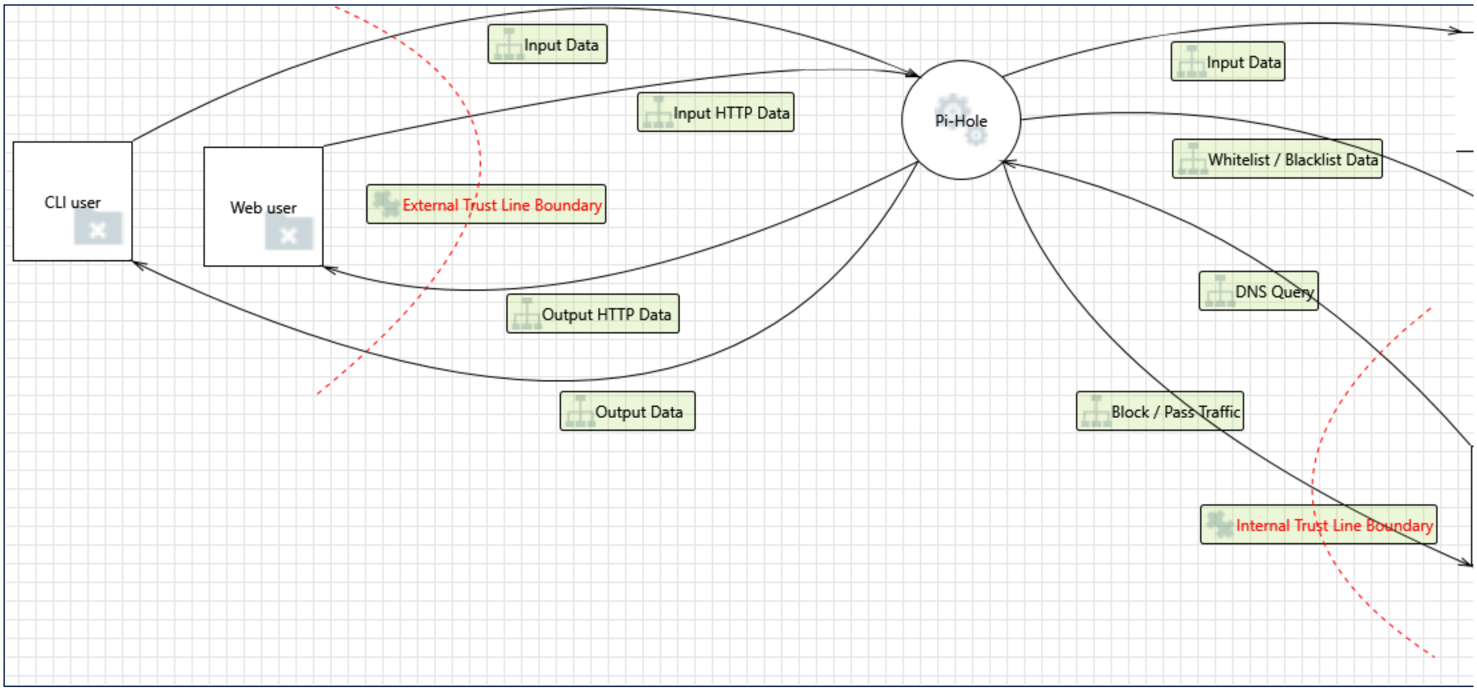
Assumptions:

External Dependencies:

Threat Model Summary:

Not Started	6
Not Applicable	0
Needs Investigation	7
Mitigation Implemented	33
Total	46
Total Migrated	0

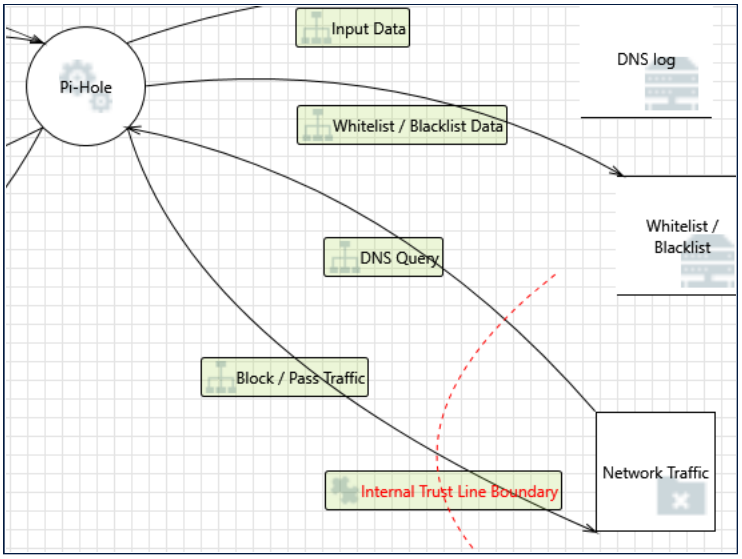
Diagram: Output Data



Output Data Diagram Summary:

Not Started	6
Not Applicable	0
Needs Investigation	7
Mitigation Implemented	33
Total	46
Total Migrated	0

Interaction: Block / Pass Traffic



1. Spoofing of the Network Traffic External Destination Entity [State: Not Started] [Priority: High]

Category: Spoofing

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

Justification: All traffic uses internal processes, physical access is controlled by user.

2. External Entity Network Traffic Potentially Denies Receiving Data [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Network Traffic 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: All the traffic data is stored in internal traffic logs.

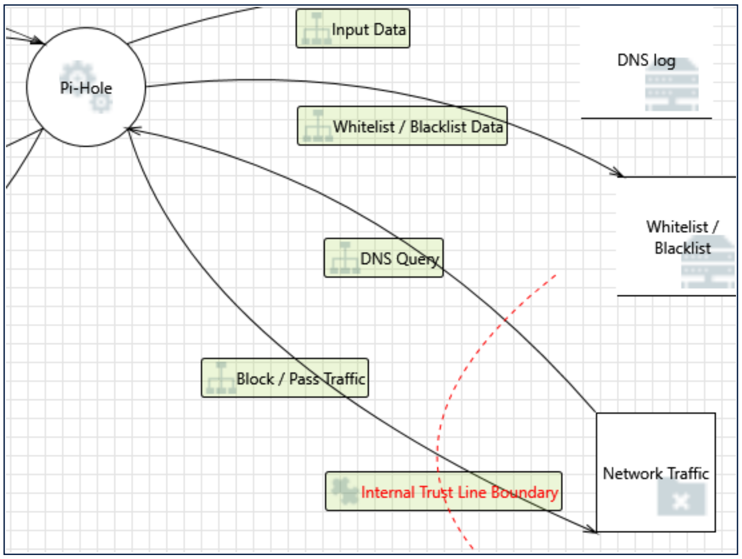
3. Data Flow Block / Pass Traffic Is Potentially Interrupted [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

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

Justification: All the traffic is on internal network, physical access is assumed controlled by user.

Interaction: DNS Query



4. Spoofing the Pi-hole Process [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Pi-hole may be spoofed by an attacker and this may lead to information disclosure by Network Traffic. Consider using a standard authentication mechanism to identify the destination process.

Justification: Pi-Hole implements DNSSec to mitigate the spoofing.

5. Spoofing the Network Traffic External Entity [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Network Traffic may be spoofed by an attacker and this may lead to unauthorized access to Pi-hole. Consider using a standard authentication mechanism to identify the external entity.

Justification: All traffic uses internal processes, physical access is controlled by user.

6. Potential Lack of Input Validation for Pi-hole [State: Needs Investigation] [Priority: High]

Category: Tampering

Description: Data flowing across DNS Query may be tampered with by an attacker. This may lead to a denial of service attack against Pi-Hole or an elevation of privilege attack against Pi-Hole or an information disclosure by Pi-Hole. 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: Data flowing across DNS Query should be encrypted using standard encryption methods.

7. Potential Data Repudiation by Pi-hole [State: Not Started] [Priority: High]

Category: Repudiation

Description: Pi-Hole 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: All the traffic data is stored in internal traffic logs.

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

Category: Information Disclosure

Description: Data flowing across DNS Query 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: All traffic data is on internal network, physical access is assumed controlled by user.

9. Potential Process Crash or Stop for Pi-hole [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

Description: Pi-Hole crashes, halts, stops or runs slowly; in all cases violating an availability metric.

Justification: Pi-Hole process relies on internal OS to monitor health and provide restart of process when needed.

10. Data Flow DNS Query 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: All the traffic is on internal network, physical access is assumed controlled by user.

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

Category: Elevation Of Privilege

Description: Pi-Hole may be able to impersonate the context of Network Traffic in order to gain additional privilege.

Justification: Pi-Hole process only sends IP address to be searched, no additional privilige identified.

12. Pi-hole May be Subject to Elevation of Privilege Using Remote Code Execution [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: Network Traffic may be able to remotely execute code for Pi-Hole.

Justification: Pi-Hole process only utilizes DNS traffic.

13. Elevation by Changing the Execution Flow in Pi-hole [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

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

Justification: Pi-Hole process utilizes scripts to check against send DNS requests

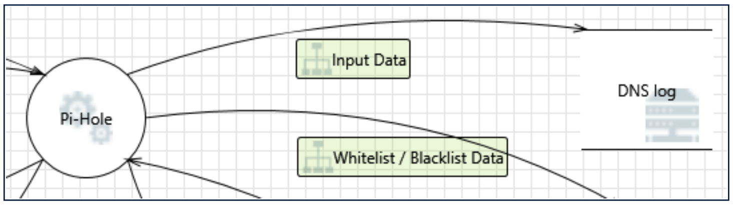
14. Cross Site Request Forgery [State: Mitigation Implemented] [Priority: High]

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: Pi-Hole process utilizes local intranet pages to conduct HTTP requests to the Pi-Hole process.

Interaction: Input Data



15. Spoofing of Destination Data Store DNS log [State: Not Started] [Priority: High]

Category: Spoofing

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

Justification: DNS log file should be authenticated using hashes or digital signatures. This needs to be implemented by Pi-Hole.

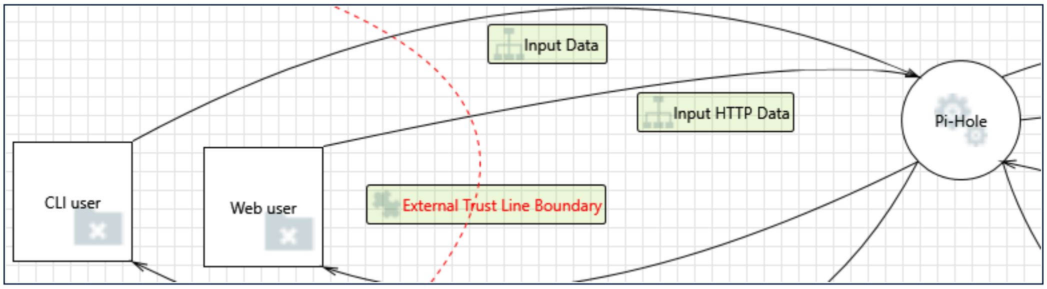
16. Potential Excessive Resource Consumption for Pi-hole or DNS log [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

Description: Does Pi-Hole or DNS log 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: DNS log is a script file controlled by Pi-Hole process.

Interaction: Input Data



17. Spoofing the Pi-hole Process [State: Not Started] [Priority: High]

Category: Spoofing

Description: Pi-hole may be spoofed by an attacker and this may lead to information disclosure by CLI user. Consider using a standard authentication mechanism to identify the destination process.

Justification: CLI user are authenticated using their Pi-Hole account password

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

Category: Spoofing

Description: CLI user may be spoofed by an attacker and this may lead to unauthorized access to Pi-hole. Consider using a standard authentication mechanism to identify the external entity.

Justification: Standard Authentication exists in Pi-Hole. SSH is required for access over network, the user password is also required to run Pi-Hole commands.

19. Potential Lack of Input Validation for Pi-hole [State: Needs Investigation] [Priority: High]

Category: Tampering

Description: Data flowing across Input Data Flow may be tampered with by an attacker. This may lead to a denial of service attack against Pi-Hole or an elevation of privilege attack against Pi-Hole or an information disclosure by Pi-Hole. 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: Data flowing across Input Data flow should be encrypted using standard encryption methods.

20. Potential Data Repudiation by Pi-hole [State: Not Started] [Priority: High]

Category: Repudiation

Description: Pi-Hole 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: All the traffic data is stored in internal traffic logs.

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

Category: Information Disclosure

Description: Data flowing across Input Data Flow 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: All traffic data is on internal network, physical access is assumed controlled by user.

22. Potential Process Crash or Stop for Pi-hole [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

Description: Pi-Hole crashes, halts, stops or runs slowly; in all cases violating an availability metric.

Justification: Pi-Hole process relies on internal OS to monitor health and provide restart of process when needed.

23. Data Flow Input Data Flow 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: All the traffic is on internal network, physical access is assumed controlled by user.

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

Category: Elevation Of Privilege

Description: Pi-Hole may be able to impersonate the context of CLI user in order to gain additional privilege.

Justification: CLI interface utilizes only Pi-Hole process.

25. Pi-hole May be Subject to Elevation of Privilege Using Remote Code Execution [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: CLI user may be able to remotely execute code for Pi-Hole.

Justification: Pi-Hole controls input with command line arguments.

26. Elevation by Changing the Execution Flow in Pi-hole [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

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

Justification: Pi-Hole process utilizes scripts to check against send DNS requests

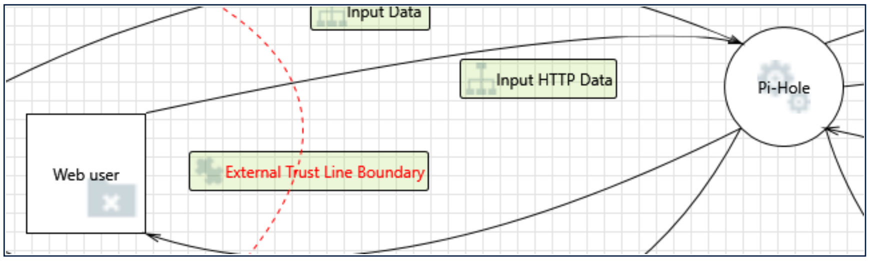
27. Cross Site Request Forgery [State: Mitigation Implemented] [Priority: High]

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: Pi-Hole process utilizes local intranet pages to conduct HTTP requests to the Pi-Hole process.

Interaction: Input HTTP Data



28. Spoofing the User External Entity [State: Needs Investigation] [Priority: High]

Category: Spoofing

Description: Web user may be spoofed by an attacker and this may lead to unauthorized access to Pi-hole. Consider using a standard authentication mechanism to identify the external entity.

Justification: Authentication mechanism is used but needs to be well implemented.

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

Category: Elevation Of Privilege

Description: Pi-Hole may be able to impersonate the context of Web user in order to gain additional privilege.

Justification: Pi-Hole web interface utilizes only Pi-Hole process.

30. Spoofing the Pi-hole Process [State: Needs Investigation] [Priority: High]

Category: Spoofing

Description: Pi-hole may be spoofed by an attacker and this may lead to information disclosure by Web user. Consider using a standard authentication mechanism to identify the destination process.

Justification: Authentication mechanism is used but needs to be implemented in a better way. IPSec could be used to authenticate traffic.

31. Potential Lack of Input Validation for Pi-hole [State: Needs Investigation] [Priority: High]

Category: Tampering

Description: Data flowing across Input HTTP Data may be tampered with by an attacker. This may lead to a denial of service attack against Pi-Hole or an elevation of privilege attack against Pi-Hole or an information disclosure by Pi-Hole. 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: Data flowing across Input HTTP data should be encrypted using standard encryption methods.

32. Potential Data Repudiation by Pi-hole [State: Not Started] [Priority: High]

Category: Repudiation

Description: Pi-Hole 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: All the traffic data is stored in internal traffic logs.

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

Category: Information Disclosure

Description: Data flowing across Input HTTP 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: All traffic data is on internal network, physical access is assumed controlled by user.

34. Potential Process Crash or Stop for Pi-hole [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

Description: Pi-Hole crashes, halts, stops or runs slowly; in all cases violating an availability metric.

Justification: Pi-Hole process relies on internal OS to monitor health and provide restart of process when needed.

35. Data Flow Input Data Flow 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: All the traffic is on internal network, physical access is assumed controlled by user.

36. Pi-hole May be Subject to Elevation of Privilege Using Remote Code Execution [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: Web user may be able to remotely execute code for Pi-Hole.

Justification: Pi-Hole controls input with command line arguments by utilizing web interface to launch scripts.

37. Elevation by Changing the Execution Flow in Pi-hole [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

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

Justification: Pi-Hole process utilizes scripts to check against send DNS requests

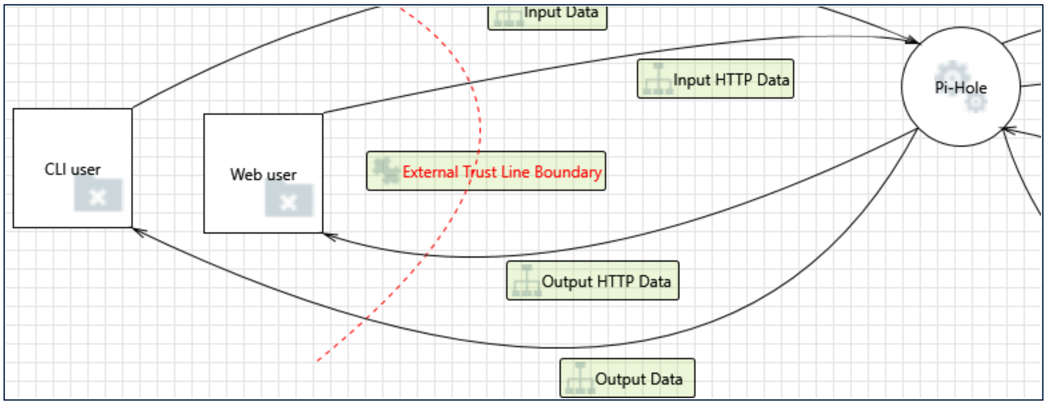
38. Cross Site Request Forgery [State: Mitigation Implemented] [Priority: High]

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: Pi-Hole process utilizes local intranet pages to conduct HTTP requests to the Pi-Hole process.

Interaction: Output Data



39. Spoofing of the CLI user External Destination Entity [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

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

Justification: Standard Authentication exists in Pi-Hole. SSH is required for access over network, the user password is also required to run Pi-Hole commands.

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

Category: Repudiation

Description: CLI 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: All the traffic data is stored in internal traffic logs.

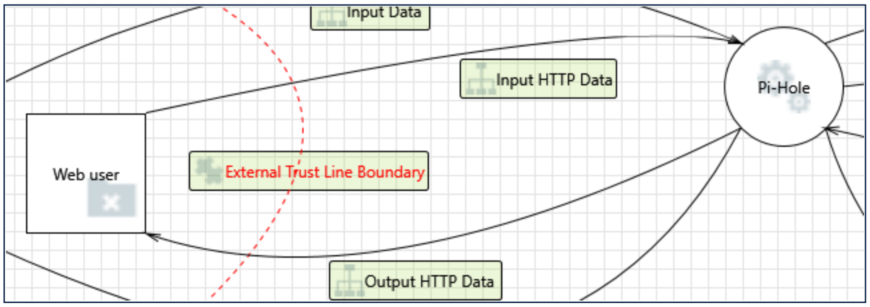
41. Data Flow Output Data Flow Is Potentially Interrupted [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

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

Justification: All the traffic is on internal network, physical access is assumed controlled by user.

Interaction: Output HTTP Data



42. Spoofing of the User External Destination Entity [State: Needs Investigation] [Priority: High]

Category: Spoofing

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

Justification: Authentication mechanism is used but needs to be well implemented. IPSec could be used to authenticate traffic.

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

Category: Repudiation

Description: Web 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: All the traffic data is stored in internal traffic logs.

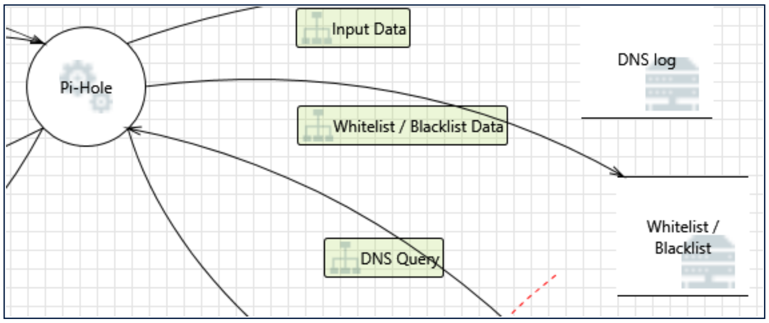
44. Data Flow Output Data Flow Is Potentially Interrupted [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

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

Justification: All the traffic is on internal network, physical access is assumed controlled by user.

Interaction: Whitelist / Blacklist Data



45. Spoofing of Destination Data Store Whitelist / Blacklist [State: Needs Investigation] [Priority: High]

Category: Spoofing

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

Justification: Whitelist / Blacklist files should be authenticated using hashes or digital signatures and use file permissions. This needs to be implemented by Pi-Hole.

46. Potential Excessive Resource Consumption for Pi-hole or Whitelist / Blacklist [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

Description: Does Pi-Hole or Whitelist / Blacklist 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: Whitelist/Blacklist is a script file controlled by Pi-Hole process.