feroot

Automate PCI DSS 4.0.1 Today

How to Automate Compliance with Requirements 6.4.3 and 11.6

This document focuses on Requirement 6.4.3 and Requirement 11.6.1. By following this guide, you will be able to automate the reporting, protection, monitoring, and response mechanisms for your payment pages.

STEP 1:
MAINTAIN AN INVENTORY OF NECESSARY SCRIPTS, ASSURE SCRIPT INTEGRITY, AND CONFIRM THAT EACH SCRIPT IS AUTHORIZED.

STEP 2:
AUTOMATE PREVENTION AND DETECTION OF UNEXPECTED SCRIPT ACTIVITIES.

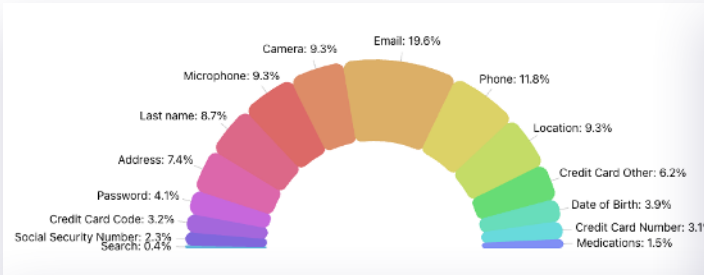
STEP 3:
ACTIVATE NOTIFICATION ALERTS FOR UNAUTHORIZED SCRIPTS AND UNEXPECTED SCRIPT ACTIVITIES.

How to Use Feroot to Comply with PCI DSS 4.0.1 Requirements 6.4.3 and 11.6.1 on Your Payment Pages.

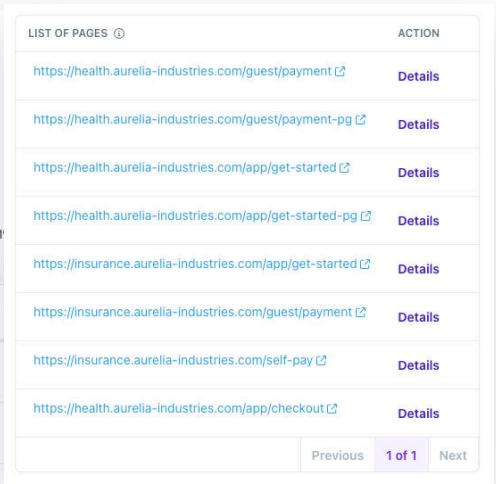
This user guide will walk you through how to use Feroot's suite of tools to meet PCI-DSS requirement on your e-commerce webpages that handle card payments.

Scoping - identify all web pages to be in scope for compliance with PCI DSS 4.0.1 Requirements 6.4.3 and 11.6.1.

- 1. Open the Data Assets report to review the inventory of web pages that collect Cardholder Data (CHD) to identify web pages and iframes that will be in scope for compliance with Requirements 6.4.3 and 11.6.1.



ASSET NAME	PRESENCE	DETECTIONS	PAGES
Credit Card Number	Input	409	8
Name	Input	674	121
Date of Birth	Device	515	27
Social Security Number	Input	176	5
Search Bar	Input	173	121
Date of Birth	Input	346	12
Password	Input, Page text	318	9
Email	Page text	95	75
Medical Conditions	Input	95	95
Social Security Number	Input, Page text	215	6



Definition: Payment Page

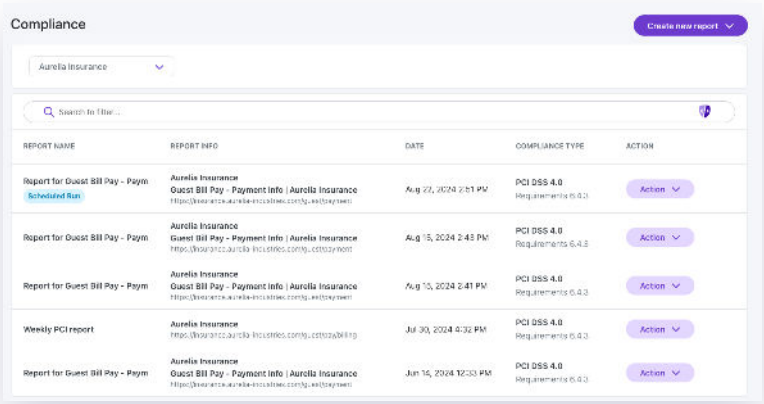
A web-based user interface containing one or more form elements intended to capture account data from a consumer or submit captured account data, for purposes of processing and authorizing payment transactions. The payment page can be rendered as any one of:

- *A single document or instance,*
- *A document or component displayed in an inline frame within a non-payment page.*

Multiple documents or components each containing one or more form elements contained in multiple inline frames within a non-payment page.

Step 1: Maintain an Inventory of Necessary Scripts, assure the integrity of each script and confirm that each script is authorized.

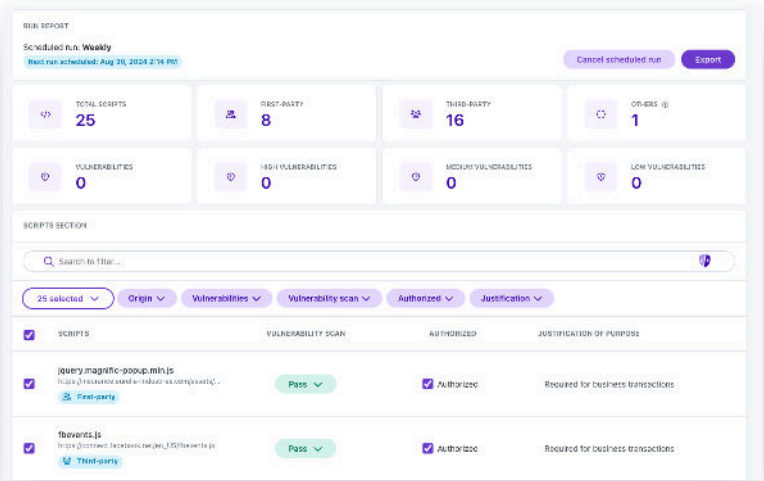
- 1. Navigate to the PCI-DSS Compliance section
- 2. Review your existing reports or
- 3. Click **Create new report** button and select the payment webpage



Definition:

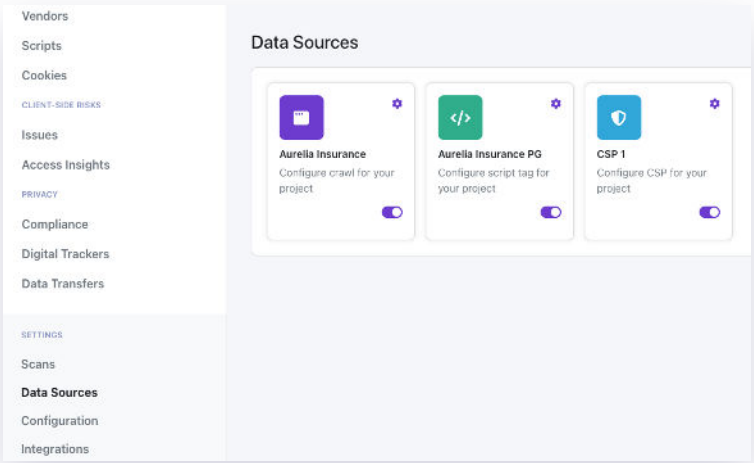
"Necessary" for PCI DSS 4.0.1 means: *necessary for the operation of the payment page. This replaced PCI DSS 4.0's definition which was "…needed for the functionality of the payment page to accept a payment transaction".*

- 4. Review and edit Justification and Authorization status of all scripts
- 5. Select your desired schedule for this report and click on **Save** button



Step 2: Automate prevention and detection of unexpected script activities.

- 1. Open **Data Sources** and activate DomainGuard Content Security Policy (CSP) and/or PageGuard Tag.
- 2. Activate JS tag-management security policy with **PageGuard** to prevent malicious script execution by permitting only Authorized and Justified scripts on payment page.

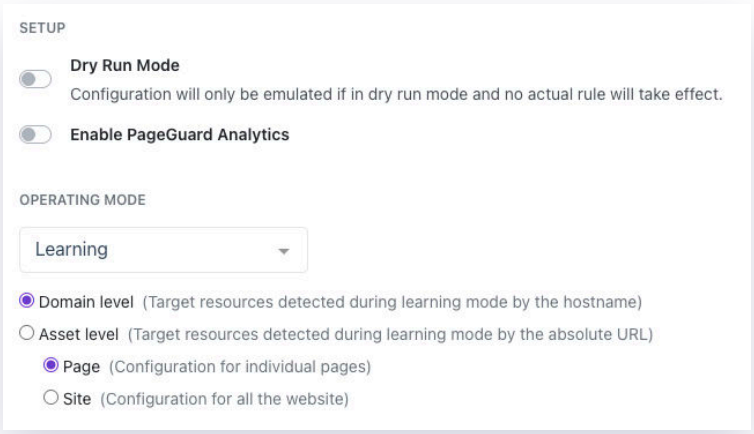


Applicability Notes:

The intention of requirement 11.6.1 is not that an entity installs software in the systems or browsers of its consumers, but rather that the entity uses techniques such as those described under Examples to prevent and detect unexpected script activities.

This requirement applies to all scripts loaded from the entity's environment and scripts loaded from third and fourth parties. This requirement also applies to scripts in the entity's webpage(s) that includes a TPSP's/ payment processor's embedded payment page/form (for example, one or more inline frames or iframes).

- 3. Open **PageGuard** to set your desired compliance and security policy.



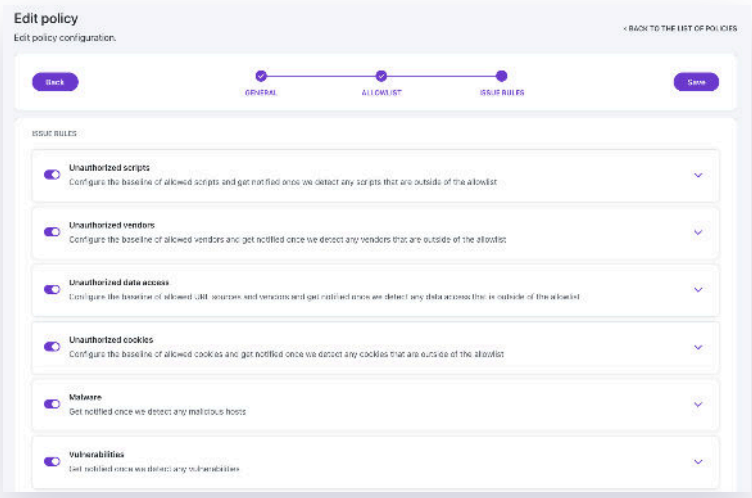
Examples

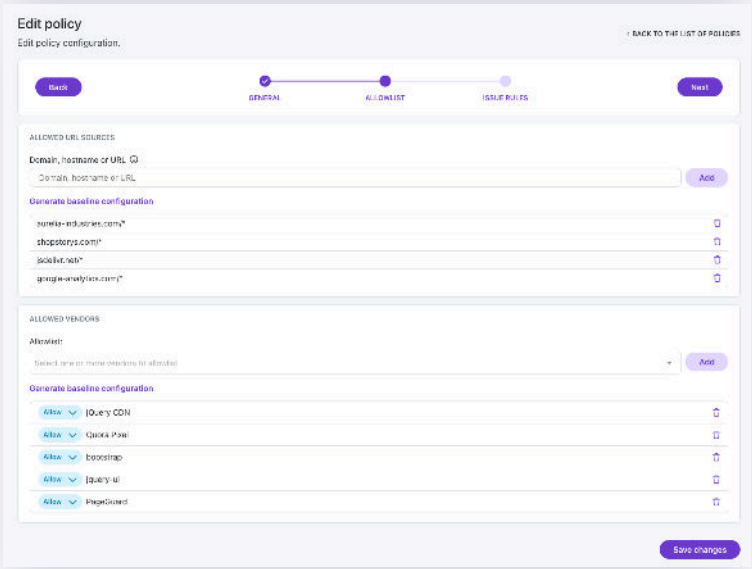
Mechanisms that detect and report on changes to the headers and content of the payment page could include, but are not limited to:

- *External monitoring by systems that request and analyze the received web pages (also known as synthetic user monitoring) can detect changes to JavaScript in payment pages and alert personnel.*
- *Embedding tamper-resistant, tamper-detection script in the payment page can alert and block when malicious script behavior is detected.*

Step 3: Activate Notification Alerts for Unauthorized Scripts and unexpected script activities.

- 1. Open existing or create new Notification Policy in the **Policies** to activate Notification Alerts.
- 2. Select desired notification rules and create the baseline of what is allowed.
- 3. You will receive notification alerts via email, Slack/Teams, PagerDuty or other systems of your choice.





Best Practices and Tips

• Understand Script Functionality: Regularly review the functionality of all scripts on your payment page to ensure they are necessary for its operation.
• Monitor for Unauthorized Script Behavior: Use Feroot's tools to monitor scripts for any unauthorized behavior, such as data skimming or other malicious activities.
• *Understand Your Content: Be aware of all external and internal scripts running on your payment pages. Regularly audit and justify their necessity.*
• *Stay Informed: Keep up-to-date with the latest security threats and ensure your monitoring tools are configured to detect these new threats.*
• *Educate Your Team: Ensure that all team members involved in website maintenance and security are aware of the importance of PCI DSS compliance and know how to respond to alerts.*

By following these steps and utilizing Feroot's comprehensive security solutions, you can ensure that your payment pages are not only compliant with PCI DSS 4.0 requirement 6.4.3 but also offer a secure environment for your customers to conduct transactions.

Integrate With Your Tools

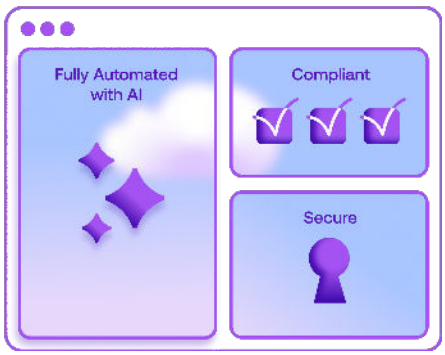
Overview of Data Collection Performed by Feroot Inspector and Feroot PageGuard:

Inspector

The Feroot Inspector performs Deep Discovery without compromising user privacy. It operates as an external crawler (scanner) that simulates typical user journeys, generating and utilizing only synthetic data during its testing processes. Therefore, the Inspector does not access any actual user information. As a result, the Inspector remains entirely isolated from user data or online visitor information, ensuring that it does not collect or process any real user data.

The Feroot Inspector **does not** collect sensitive user information, including Personally Identifiable Information (PII), Protected Health Information (PHI), Cardholder Data (CHD), user session information, or any other user-related data.

PageGuard



Feroot PageGuard is a script-tag solution deployed on webpages to deliver real-time compliance and security protection. It captures only a sample of IP addresses from incoming requests and generates derivative data for threat and compliance intelligence purposes. This derived information is anonymized and excludes any personal user information or other identifiable data, ensuring that user privacy remains intact while providing effective compliance and security insights and protection.

Feroot PageGuard **does not** collect or access sensitive information, such as Personally Identifiable Information (PII), Protected Health Information (PHI), Cardholder Data (CHD), user session information, or any other user-related data.