Payment Card Industry Data Security Standard (PCI DSS) v4.0.1

Schellman Review: Fastly Client-Side Protection

fastly + schellman

Table of Contents

Section 1: Executive Summary

Purpose

Schellman Compliance, LLC (Schellman) assessed Fastly, Inc.'s (Fastly) Client-Side Protection solution to evaluate its capabilities in helping organizations meet applicable requirements of the Payment Card Industry Data Security Standard (PCI DSS) v4.0.1, with particular focus on Requirements 6.4.3 and 11.6.1.

Schellman examined Fastly's Client-Side Protection product capabilities across several domains: script inventory management, Content Security Policy implementation, change detection mechanisms, and compliance reporting functions. The review assessed the features against specific PCI DSS requirements and identified control areas where compliance challenges are common.

Scope of Work & Approach Taken

Schellman's review included technical documentation analysis, interviews with Fastly product specialists, and an examination of the platform's functionality, including Client-Side Protection product features, implementation requirements, and operational capabilities to determine their alignment with PCI DSS compliance requirements.

Company Background & Services Provided

Fastly is a cloud computing services provider specializing in content delivery network (CDN) services, edge cloud platforms, and security solutions. The Client-Side Protection product inventories the resources that the user's browser loads from the Fastly customer's site as delivered through the Edge Cloud Platform and then controls those resources by generating and enforcing content security policies.

Review Summary

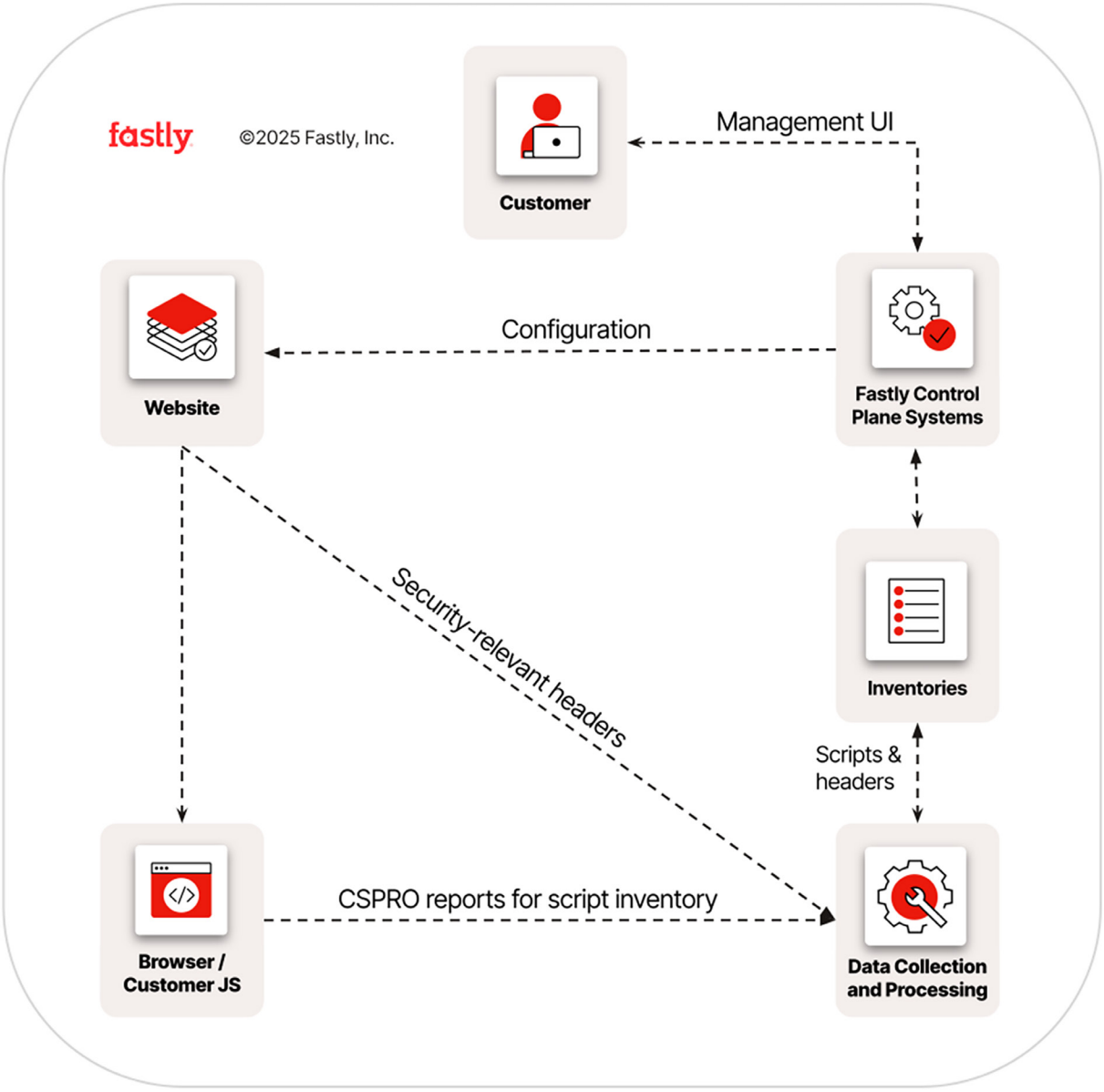
During Schellman's review of Fastly Client-Side Protection, the following observations were noted:

- Fastly Client-Side Protection directly assists organizations in meeting the requirements in Requirement 6.4.3 for managing client-side scripts used on parent pages and payment pages.
- The policy creation and enforcement functionality provided by Client-Side Protection assists with Requirement 6.4.3 by maintaining an inventory of authorized script sources.
- The product includes mechanisms for detecting, inventorying, and controlling client-side scripts, addressing elements of Requirements 6.4.3 and 11.6.1.
- The product provides the necessary tools to assist organizations in effectively mitigating cross-site scripting attacks (e.g., Magecart attacks) that target payment information.
- Fastly documentation for Client-Side Protection explains the customer's options and responsibilities for the appropriate configuration of the tool.

Section 2: Client-Side Protection Methodology

Product Architecture

Fastly Client-Side Protection operates in conjunction with Fastly's Next-Gen Web Application Firewall (WAF) to inventory and control resources loaded on the user's browser. The product works by modifying response headers to implement content security policies and collect information about client-side scripts executing in user browsers.



The core functionality supports inserting and managing one of two headers: Content-Security-Policy for enforcement (blocking mode) or Content-Security-Policy-Report-Only for monitoring (logging mode). This approach enables both monitoring (logging mode) and enforcement (blocking mode) of content security policies based on organizational requirements.

Deployment Options

Client Side Protection supports multiple deployment configurations through the Next-Gen WAF, Edge WAF, and Cloud WAF. Schellman's review confirmed that the following WAF deployment variations support the Client-Side Protection product:

1. **Apache Module (agent version 1.10.0+)**
2. **AWS Lambda (agent version 4.64.0+)**
3. **Envoy (agent version 4.62.0+)**
4. **Golang (agent version 4.61.0+)**
5. **HAProxy (agent version 4.63.0+, module version 1.5.0+)**
6. **Heroku (agent version 4.61.0+)**
7. **IBM Cloud (agent version 4.61.0+)**
8. **IIS Module (agent version 3.5.0+)**
9. **Java Module (agent version 2.7.0+)**
10. **.NET Module (agent version 1.7.1+)**
11. **.NET Core Module (agent version 1.4.1+)**
12. **NGINX Lua Module (agent version 1.7.0+)**
13. **NGINX Native Module (agent version 1.2.0+)**
14. **Node.js Module (agent version 2.3.0+)**
15. **Reverse Proxy (agent version 4.61.0+)**
16. **VMware Tanzu (agent version 4.61.0+)**

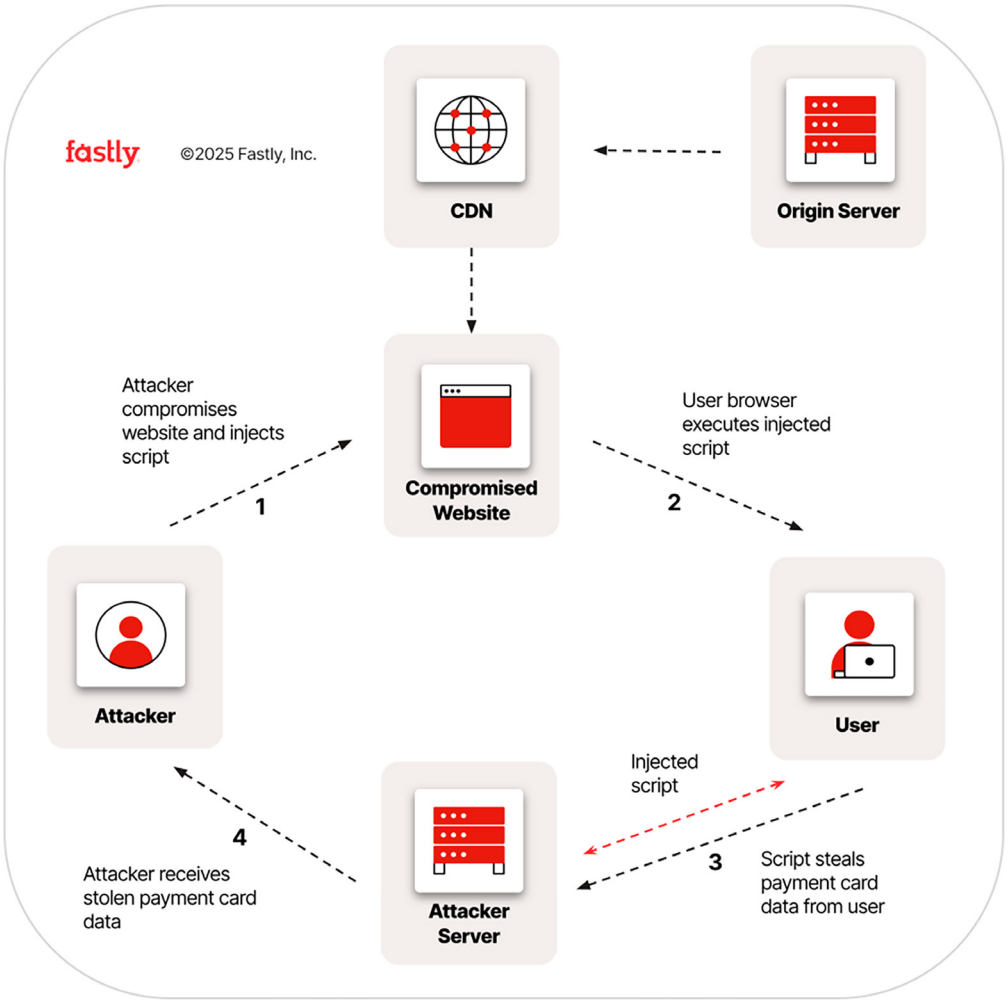
Note that PHP and Python deployment variations do not currently support Client-Side Protection.

Operational Workflow

The Client-Side Protection product follows a methodology that aligns with current security best practices for web application protection endorsed by organizations such as NIST, OWASP, SCAP, and the Mozilla Foundation. The operational workflow consists of:

1. **Definition of scope through website and page configurations**
2. **Script inventory collection and monitoring**
3. **Policy creation and management**
4. **Enforcement and violation reporting**

These workflow elements form the core capabilities of Client-Side Protection as outlined in the diagram below.



Website & Page Definition

Client-Side Protection provides a framework for defining the areas of web applications to be monitored and protected. This is accomplished through the creation of websites and pages:

- A website is defined as a base URL (e.g., *https://www.example.com*)
- A page is defined as a single path (e.g., /checkout) or a collection of paths that describe a similar component or user experience (e.g., /checkout and /edit-cc)

Together, these definitions create the full URLs (e.g., *https://www.example.com/checkout*) that constitute the scope of protection. This granular approach allows organizations to focus protection efforts on critical areas of their applications, such as payment pages, which can help support PCI DSS compliance.

Script Inventory Management

Client-Side Protection creates and maintains an inventory of client-side scripts for each defined page—a mechanism that is central to its security approach. For each page identified by the customer, the service continuously updates a list of client-side scripts and security-related response headers by injecting the Content-Security-Policy-Report-Only header into a sample of responses. This enables Client-Side Protection to observe resource loading without disrupting normal operations. Scripts are added to the inventory every three days, while a housekeeping mechanism removes scripts not seen for 90 days.

The system also keeps tabs on security-related response headers for all responses carrying that reporting header. The inventory process includes:

- Collection of script information through the Content-Security-Policy-Report-Only header
- Regular sampling and updating of the inventory
- Storage of script information for 90 days, with automatic removal of scripts not detected during that time
- Classification capabilities to add authorization status and comments for each observed script

The inventory gives organizations visibility into the scripts executing on their web pages, enabling them to identify unauthorized changes to scripts or unauthorized or potentially malicious scripts. This capability supports compliance with Requirement 6.4.3, which requires maintaining an inventory of authorized scripts and detecting unauthorized modification of script contents of payment pages and security-impacting HTTP headers.

Script Inventory Management

Client-Side Protection creates and maintains an inventory of client-side scripts for each defined page—a mechanism that is central to its security approach. For each page identified by the customer, the service continuously updates a list of client-side scripts and security-related response headers by injecting the Content-Security-Policy-Report-Only header into a sample of responses. This enables Client-Side Protection to observe resource loading without disrupting normal operations. Scripts are added to the inventory every three days, while a housekeeping mechanism removes scripts not seen for 90 days.

The system also keeps tabs on security-related response headers for all responses carrying that reporting header. The inventory process includes:

- Collection of script information through the Content-Security-Policy-Report-Only header
- Regular sampling and updating of the inventory
- Storage of script information for 90 days, with automatic removal of scripts not detected during that time
- Classification capabilities to add authorization status and comments for each observed script

The inventory gives organizations visibility into the scripts executing on their web pages, enabling them to identify unauthorized changes to scripts or unauthorized or potentially malicious scripts. This capability supports compliance with Requirement 6.4.3, which requires maintaining an inventory of authorized scripts and detecting unauthorized modification of script contents of payment pages and security-impacting HTTP headers.

Content Security Policy Management

Client-Side Protection includes tools for creating and managing content security policies that control the resources loaded in the user's browser. Each policy created applies to a specific page and consists of directives controlling different resource types. (The script-src directive, for instance, determines which JavaScript sources are permitted.)

The policy injection mechanism is straightforward—the Next-Gen WAF adds the specified policies to responses via either the Content-Security-Policy header (in full blocking mode) or the Content-Security-Policy-Report-Only header (in monitoring mode). This approach delivers policies directly to the browser, which then handles enforcement or monitoring according to customer settings.

The policy management capabilities include:

- Creation of policies applied to specific pages
- Definition of directives that control different types of resources (e.g., for JavaScript)
- Configuration of allowed sources for each directive type
- Selection of protection modes (blocking, logging, or off)
- Activation workflow for policy changes

The policy management functionality enables Fastly customer organizations to implement the controls required by Requirement 6.4.3, which mandates managing client-side scripts used on payment pages.

Violation Reporting & Monitoring

Client-Side Protection provides reporting and monitoring capabilities that give organizations visibility into policy violations. At regular intervals, Fastly e-mails notifications of response header changes to the designated contact, grouping the changes into hourly intervals for review. When script changes are detected, the system automatically resets their authorization status, requiring a human review for re-approval. This process ensures that script changes are detected for re-evaluation.

Schellman verified that the reporting system:

- Collects and stores unique policy violations for 90 days
- Increases counts for repeat violations
- Provides violation details, including directive, blocked addresses, and HTTP status code
- Facilitates policy refinement based on violation reports

These monitoring capabilities are essential for Fastly customer organizations to maintain an effective Content Security Policy implementation and demonstrate ongoing compliance with Requirements 6.4.3 and 11.6.1.

Section 3: PCI DSS v4.0.1 Requirement Coverage

Schellman's assessment evaluated Fastly Client-Side Protection's ability to help organizations meet specific PCI DSS v4.0.1 requirements. The following sections detail Schellman's findings regarding the product's coverage of relevant requirements.

Requirement 6.4.3 Coverage

Fastly Client-Side Protection supports compliance with Requirement 6.4.3 by automatically building and maintaining the script inventory for specified pages. The inventory interface provides security teams with a method to review all detected scripts, assign approval status, and document business justifications for each script. This process provides auditors with the necessary evidence to verify that scripts running in the in-scope environment have been vetted by the assessed entity.

The inventory process also actively tracks changes and alerts customers when new or modified scripts appear. When a script changes, its approval status is reset, forcing review and reapproval. This forced re-evaluation ensures that every script change gets scrutinized and maintains the integrity of the approval process. The search and filtering capabilities can be particularly useful for managing large script inventories, allowing for isolation of unauthorized or newly detected scripts.

Requirement 6.4.3 states: "An inventory of all scripts is maintained with written justification as to why each script is necessary." Fastly Client-Side Protection supports compliance with this requirement through the following capabilities:

1. **Script Inventory Creation:** The product automatically creates and maintains an inventory of all client-side scripts executing on defined pages, supporting compliance with this requirement.

2. **Script Authorization and Justification:** The inventory functionality allows organizations to assign an authorization status (authorized or not authorized) and add justification comments for each script, supporting the requirement for written justification.

3. **Role-Based Authorization:** Only customer users with the Engineer role are permitted to authorize scripts, supporting controlled and secure management of script approvals.

4. **Change Detection and Notification:** The product monitors script changes and alerts organizations when scripts are modified, enabling timely review and re-authorization as needed.

5. **E-mail Notifications:** The system sends notifications about script changes and security-impacting response headers, supporting ongoing script inventory management.

Organizations using Client-Side Protection must ensure that they review all scripts in the inventory and document justifications for authorized scripts to fully comply with this requirement. While the tool provides the necessary capabilities, timely reviews and compliance with the PCI DSS requirements depend on each organization's implementation and ongoing maintenance of the script inventory.

Requirement 11.6.1 Coverage

Requirement 11.6.1 mandates the deployment of a change and tamper detection mechanism designed to alert personnel to any unauthorized modifications, including indicators of compromise, changes, additions, or deletions to security-impacting HTTP headers and script contents on payment pages as received by the consumer's browser. This mechanism must be configured to evaluate both the received HTTP headers and the payment pages themselves. These detection functions must be performed at least weekly or at a frequency defined in the entity's Targeted Risk Analysis (TRA), provided that the analysis is conducted by the customer organization in accordance with all elements specified in Requirement 12.3.1.

Fastly addresses this through content security policy management, which enables customers to define which sources are permitted to deliver scripts and other resources to web pages. By enforcing these policies, organizations can prevent unauthorized code execution. The policy builder interface makes the creation and management of these rules straightforward, allowing for granular controls tailored to a specific environment.

Fastly Client-Side Protection provides the following capabilities to help support this requirement:

1. **Script Change Detection:** The product monitors scripts on a continuous basis and identifies changes to existing scripts, providing the change-detection mechanism mandated by the requirement.

2. **Content Security Policy Enforcement:** Through the blocking protection mode, the product can prevent unauthorized scripts from executing, thereby protecting against unauthorized modifications that could lead to data theft.

3. **Violation Reporting:** The system logs and reports attempts to load unauthorized resources, facilitating detection of potential attacks targeting payment data—these logs are available in the Fastly control panel.

4. **Response Header Monitoring:** The product monitors and logs security-impacting response headers, which provides an additional layer of detection for potential security issues.

The combination of these capabilities supports the detection and prevention of attacks that may target payment data through unauthorized script modifications.

Additional Relevant Requirements

While Schellman's assessment focused primarily on Requirements 6.4.3 and 11.6.1, we identified additional PCI DSS v4.0.1 requirements that are supported by Client-Side Protection:

Requirement 6.4.2: "For public-facing web applications, an automated technical solution is deployed that continually detects and prevents web-based attacks, with at least the following: Is installed in front of public-facing web applications and is configured to detect and prevent web-based attacks."

Client-Side Protection, when deployed in blocking mode, serves as an automated technical solution that continually detects and prevents specific types of web-based attacks, particularly those involving unauthorized script execution. While this alone may not fully satisfy Requirement 6.4.2, it provides a valuable component of a comprehensive web application security strategy.

Requirement 10.2.1.2: "Audit logs capture all actions taken by any individual with administrative access, including any interactive user of application or system accounts."

While not a primary focus of the product or this evaluation, Client-Side Protection's administrative activities are logged within the Fastly control panel, contributing to the audit log requirements for administrative actions.

Section 4: Implementation Considerations

Prerequisites & Dependencies

It is important to note that Client-Side Protection is not enabled by default. Customers need to contact Fastly sales to purchase and activate it. Once set up, Fastly product users can see the Client-Side Protection menu in the Fastly control panel; however, only users with Engineer privileges are permitted to view and manage the data. The Next-Gen WAF must be enabled and configured to monitor the same applications that users intend to safeguard with Client-Side Protection. Customers using Cloud WAF or Edge deployments are not required to take any additional action. Users of the Core WAF deployment must verify that the Core WAF agent and module versions are up to date and compatible. Establishing these elements early helps reduce the risk of complications later.

The setup process begins with creating at least one website definition in the system. To do so:

- Log in to the Fastly control panel, navigate to Security > Client-Side Protection > Websites, and click the Add website button.
- Enter the base URL of the target site - just the domain portion (e.g., *https://www.example.com*). Either move directly to creating a page at this point or add the website and come back to page creation later.
- After defining a website, at least one page must be created to represent either a specific path or a collection of related paths.
- Give the page a meaningful identifier and description, then associate it with the parent website.
- Enter an e-mail address for alerts.
- Decide whether to inventory all paths or only specific ones—this flexibility allows organizations to focus protection where it is most needed, such as on payment pages or other sensitive areas of the application.

For further information, see Fastly's setup guide: *Setting up Client-Side Protection | Fastly Documentation*

Meanwhile, policy creation requires a balance of security with operational needs. An overly restrictive policy can break legitimate functionality, while an overly permissive one may introduce security gaps.

The policy creation flow is designed to be straightforward:

- Starting on the Policy page, select the target page, and click Create Policy.
- Add appropriate values for each directive using fetch directive syntax, with careful attention to how each directive governs different types of resources.

The **script-src** directive controls JavaScript execution – there are several syntax options:

- "None" blocks all scripts
- "Self" allows same-origin scripts
- "Host-sources" specifies allowed domains, and nonce values enable specific script tags

After configuring the directives, the customer selects a protection mode that aligns with the implementation phase and activates the policy to apply the changes.

Limitations & Constraints

Schellman identified the following limitations and constraints that organizations should consider when implementing Client-Side Protection:

1. **Caching Considerations:** When an object evaluated by the Next-Gen WAF is cached, the policy associated with the object is also cached. If policies are updated, cached objects will not reflect the updated policy until they are removed from the cache and passed through the WAF again.
2. **Policy Deletion:** Policies cannot be deleted; however, individual values can be removed from the policy directives, which can help fine-tune coverage. Alternatively, entire policies can be set to an "Off" protection mode to prevent them from being pushed to the browser.
3. **Response Header Limitation:** The product logs security-impacting response headers only for responses that have the Content-Security-Policy-Report-Only header.

Organizations should consider these limitations when planning their implementation to ensure effective use of the product for PCI DSS compliance.

Section 5: Monitoring & Reporting

Effective monitoring of policy violations is essential for identifying and mitigating potential security threats. Fastly's Reports interface offers a detailed view of activity across protected web pages, helping teams to assess security posture with greater clarity. It highlights which directives are triggering violations, the URLs of blocked resources, associated status codes, protection modes in effect, and the frequency of repeated violations.

It is advisable for Fastly customers to establish a structured review cadence: daily during the initial deployment phase and after significant application changes, then transitioning to weekly or bi-weekly reviews for ongoing operations. Collecting violation data alone is insufficient; active analysis is necessary to identify patterns that may indicate underlying security risks and to support compliance efforts.

The reporting capabilities support both routine security monitoring and formal compliance documentation. Filtering options allow teams to focus on specific timeframes or directive categories, streamlining investigations. Each violation includes metadata such as the first and most recent observation timestamps, which aids in distinguishing between isolated incidents and recurring threats. This level of detail not only supports PCI compliance requirements for monitoring and response but can also enhance the operational efficiency of the security team.

Alert Management & Notification Strategy

Alert configurations require strategic implementation. Notification e-mails are sent to the address specified during page setup; using a dedicated distribution list instead of an individual e-mail address helps ensure consistent alert delivery, even during staffing changes or absences. Alerts include detailed summaries of response header changes, grouped by hour to support efficient review. Organizations are encouraged to define tailored response procedures aligned with the severity and potential impact of different alert types in recognition of the fact that not all alerts require the same urgency.

Incident Response Integration

To maximize the effectiveness of Fastly's Client-Side Protection, organizations should incorporate it into their established incident response frameworks. In the event of unauthorized script activity or policy violations, predefined procedures should guide the investigation and remediation efforts. These procedures may include tracing the origin of suspicious scripts, blocking malicious domains, restoring affected systems from backups, or initiating forensic analysis as necessary. Integration with existing workflows ensures prompt and coordinated responses to identified threats.

Compliance & Audit Readiness

Data generated by Client-Side Protection also fulfills key compliance requirements. The script inventory, which includes authorization status and justification, directly addresses PCI DSS Requirement 6.4.3. Similarly, policy configurations and violation logs support compliance with Requirement 11.6.1. Organizations are advised to generate and retain periodic reports or exports to maintain historical compliance records. This is particularly important in anticipation of leadership transitions, personnel changes, or updates to service provider relationships. Maintaining accessible documentation supports audit preparation and demonstrates a proactive approach to security governance.

Section 6: Use Cases & Best Practices for Client-Side Protection

E-Commerce Environments

E-commerce platforms are among the most critical and apparent use cases for Client-Side Protection, as they routinely process payment card data and are frequently targeted by threat actors. Initial protection efforts should prioritize checkout flows and payment pages by enforcing strict content security policies that limit script execution to explicitly authorized sources. Multi-step checkout processes warrant individualized configuration to ensure complete coverage across each page in the customer journey. Fastly recommends a tiered protection strategy: apply the most restrictive policies to pages handling sensitive data, while employing more permissive settings on marketing and informational content.

Financial Services Applications

As financial applications face unique challenges due to the sensitivity and volume of data processed across a wide range of user interactions and typically require protection beyond payment functionality—including in areas such as account management, fund transfers, and personal information portals—these too would benefit from Client-Side Protection. Fastly recommends implementing Client-Side Protection across all user-facing components within these platforms, starting in Logging mode to gain visibility into the existing script environment. Transitioning to Blocking mode should follow once a stable baseline is established, enhancing the security of user data while preserving application functionality.

Customer Portals

Customer portals, especially those requiring authentication, are frequent targets for credential theft and data exfiltration, and so enabling Client-Side Protection for all authenticated sections would be advantageous and important, with particular emphasis on pages displaying personal, financial, or account-related information. For portals that serve multiple functions (e.g., support, account management, and purchasing), it is beneficial to define separate configurations per functional area. This segmentation allows for granular control and more effective application of content security policies based on the sensitivity of the data involved.

Summary of Operational Best Practices

Based on Schellman's assessment, the following operational best practices for organizations implementing Client-Side Protection to meet PCI DSS requirements were identified (regardless of which use cases apply):

Phased Implementation: A phased rollout strategy is often the most effective approach. Start by inventorying scripts on the most critical pages and deploying baseline policies in Logging mode. This provides visibility into third-party and inline scripts without impacting user experience. Next, evaluate and authorize scripts as appropriate, document justifications, and fine-tune policies using observed behavior. As confidence in the configuration grows, transition policies to Blocking mode, starting with lower-risk sections before progressing to high-sensitivity areas such as payment or personal data pages. Continuous monitoring procedures should be established to maintain protection and ensure long-term compliance.

Policy Tuning: Effective policy tuning requires balancing strong security enforcement with the functional needs of the business. It is advisable to begin with moderately permissive policies and gradually increase restrictions as the tool's responses to scripts become familiar. Specificity is key when defining policy directives—avoid broad patterns such as **.thirdparty.com* unless strictly necessary, and instead specify precise script paths, such as *subdomain.thirdparty.com/scripts/*. Frequent reviews of policy violations are essential for distinguishing false positives from legitimate threats. Policy tuning should be recognized as an ongoing process that must evolve alongside application changes and updates to third-party scripts to remain effective.

Scope Definition: Organizations should carefully define the scope of protection to include all payment pages and related components. At a minimum, the scope should include:

- All pages where payment card data is entered, processed, or displayed
- Supporting pages that are part of the payment flow
- Administrative interfaces for payment-related functionality

A well-defined scope ensures that all relevant pages are monitored and protected, which is essential for PCI DSS compliance.

Authorization Workflow: A formal workflow for script authorization should be established to include:

1. Identification of organization members with script approval authority
2. Regular review of the script inventory to identify new or changed scripts
3. Assessment of each script's necessity and security implications
4. Documentation of justifications for authorized scripts
5. Implementation of policies to block unauthorized scripts

This workflow should help organizations maintain an accurate and up-to-date script inventory with appropriate justifications, as required by Requirement 6.4.3.

Policy Management: Effective management of content security policies is critical for preventing unauthorized scripts from executing. The following are considered best practices for Policy Management:

1. Start with policies in logging mode to identify legitimate sources before enabling blocking
2. Regularly review violation reports to refine policies
3. Test policy changes in non-production environments before activation
4. Implement a change management process for policy updates
5. Document the rationale for policy decisions

These practices help organizations establish and maintain effective content security policies that protect against unauthorized script execution.

Monitoring and Response: Continuous monitoring and timely response to detected issues are essential components of an effective implementation. The following are considered best practices for Monitoring and Response activities:

1. Establish procedures for reviewing e-mail notifications about script and header changes
2. Define response protocols for unauthorized script detections
3. Regularly review violation reports to identify potential security issues
4. Implement a process for investigating and remediating identified issues

These monitoring and response practices help organizations maintain ongoing compliance with PCI DSS requirements for protecting payment data.

Section 7: Conclusions

Overall, Schellman's evaluation confirmed that Fastly's Client-Side Protection offers organizations the capabilities necessary to meet PCI DSS requirements concerning client-side script security. Specifically, the product supports Requirement 6.4.3 by maintaining a detailed inventory of client-side scripts, including the authorization status of each script and accompanying justification records. Client-Side Protection also supports Requirement 11.6.1 through robust policy enforcement features designed to detect and prevent client-side skimming and similar attack vectors.

These combined features not only bolster an organization's ability to safeguard user payment data against sophisticated browser-based threats but also provide the documentation and operational transparency required to satisfy audit standards.

When evaluating a product to help meet PCI application security requirements, ease of use, granular control over delegated functions, and implementation visibility are crucial. Client-Side Protection offers granular visibility into the scripts and resources executing within the user's browser, enabling security teams to monitor, assess, and respond to unexpected changes. The integrated change detection mechanism facilitates rapid identification of anomalies, and the policy enforcement engine blocks unauthorized code execution, thereby reducing the risk of compromise.

These capabilities can contribute to the satisfaction of PCI DSS payment script management requirements.

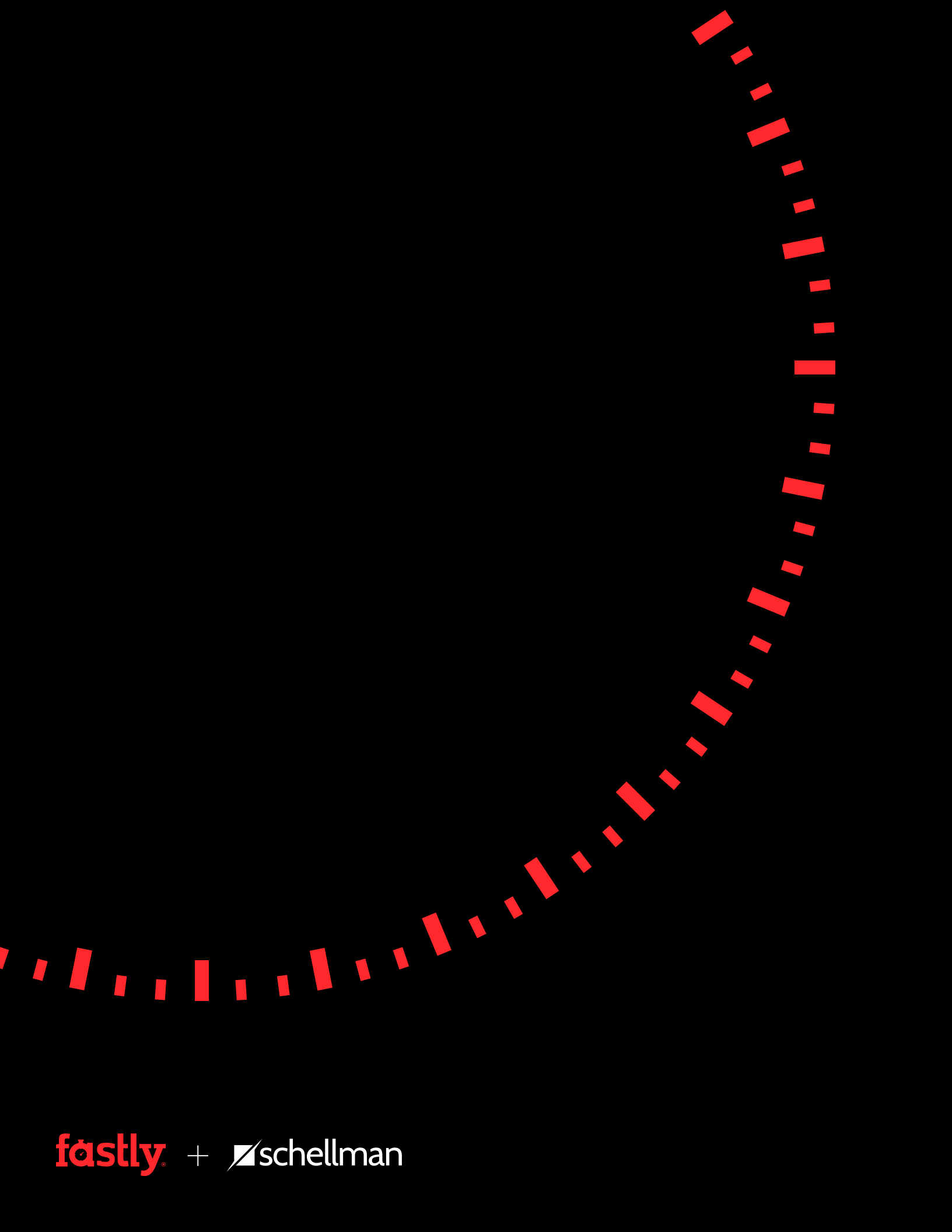
Implementation Considerations

Organizations considering the deployment of Client-Side Protection should conduct a thorough assessment of their current security posture, compliance obligations, and operational constraints. It's important to note that Fastly Client-Side Protection is built upon the Fastly Next-Gen Web Application Firewall (WAF), which must be deployed as a foundational component.

The product is intended to be integrated into a broader defense-in-depth strategy, which may include network segmentation, patch and vulnerability management, and formalized incident response processes. While the policy enforcement features are effective when properly designed and implemented, they can introduce operational complexity if not carefully managed. To maintain a balance between strong security enforcement and minimal business disruption, organizations are expected to allocate sufficient time for testing, tuning, and phased deployment.

Appendix A: Glossary of Terms

- **Client-Side Protection:** Fastly's product for inventorying and controlling resources that load on the user's browser.
- **Content Security Policy (CSP):** A security standard that helps prevent various types of attacks, including Cross-Site Scripting (XSS) and data injection attacks.
- **Content-Security-Policy Header:** A response header that allows website administrators to control resources the user agent is allowed to load for a given page.
- **Content-Security-Policy-Report-Only Header:** A response header that allows monitoring of policy violations without enforcing the policy.
- **Cross-Site Scripting (XSS):** A type of security vulnerability through which attackers can inject malicious client-side scripts into web pages that are then executed in the browsers of other users.
- **Customer:** A user of Fastly products, such as Client-Side Protection.
- **Directive:** Part of a content security policy that controls a specific type of resource (e.g., script-src for JavaScript).
- **Magecart Attack:** A form of web skimming attack where threat actors inject malicious JavaScript, often through vulnerable third-party components, into e-commerce websites to intercept and steal user payment information during checkout.
- **Next-Gen WAF:** Fastly's Web Application Firewall that inspects web traffic and helps protect web applications from a wide range of attacks, including SQL injection, cross-site scripting, and other common threats.
- **Parent Page:** A web page that hosts a payment page, whether inline or as one or more embedded iFrame or forms.
- **Payment Page:** A web-based user interface containing one or more form elements intended to capture Cardholder Data (CHD).
- **Payment Page Scripts:** Any programming language commands or instructions on a payment page that are processed and/or interpreted by a consumer's browser, including commands or instructions that interact with a page's document object model.
- **PCI DSS:** The Payment Card Industry Data Security Standard is a set of security requirements designed to ensure that any entity involved in the processing, storage, transmission, or protection of cardholder data maintains a secure environment.
- **Protection Mode:** The enforcement level of a content security policy (blocking, logging, or off).
- **Script Inventory:** A collection of client-side scripts observed on a web page, including information about each script's authorization status and justification.
- **User:** The end user of software provided by Fastly customers (see customer above). Also, the user of Fastly Client-Side Protection, where context indicates an individual operating the product.


fastly
+
schellman