DOMDOG

The Ultimate Guide for 6.4.3 & 11.6.1 PCI DSS 4.0.1 Requirements Compliance

The What? Why? How?

You Need to Know to Make an Informed Decision

The Ultimate Guide for

6.4.3 & 11.6.1

PCI DSS 4.0.1 Requirements Compliance

The What? Why? How? — You Need to Know to Make an Informed Decision

Lavakumar Kuppan & Sukesh Pappu

Contents

Preface

PCI DSS 4.0 brings two new controls—6.4.3 and 11.6.1—and you're likely determining the most effective strategies to ensure compliance. Whether you're assessing market-leading products, exploring advanced technical approaches, or considering custom solutions suggested by your internal teams, this handbook is just what you need!

While there's a wealth of information available through webinars, blog posts, and discussions on forums such as Reddit, this expert-created handbook is designed to provide you with the insights needed to make the most informed choice based on your unique needs and use case.

Lavakumar Kuppan
Founder

Sukesh Pappu
Co-founder

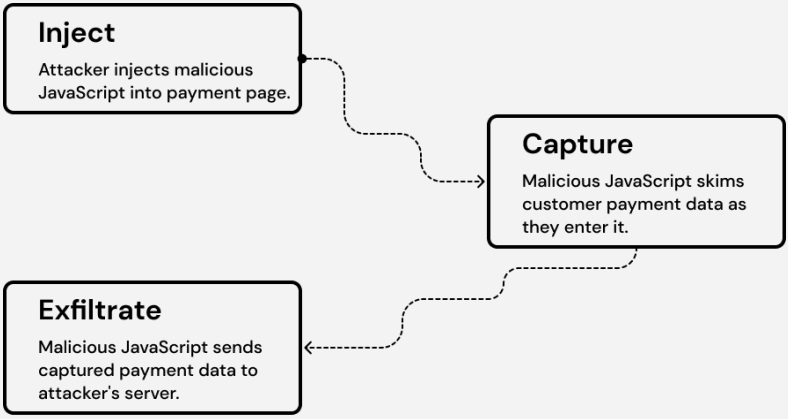
DOMDOG
Page Security & Privacy Platform

https://domdog.io

Why 6.4.3 & 11.6.1 Requirements Were Created

As digital transactions continue to rise. Web skimming attacks have become a critical threat to payment page security. These attacks target the point of data collection, where attackers inject malicious JavaScript into payment pages to capture sensitive information, such as credit card numbers, which are then transmitted to an attacker-controlled server.

The increasing number of breaches and the growing sophistication of these advanced threats have highlighted the limitations of existing controls under PCI DSS 3.2.1. To address these vulnerabilities, PCI DSS 4.0 introduced new, targeted controls—specifically Sections 6.4.3 and 11.6.1—designed to enhance payment page security and mitigate the risks posed by these increasingly sophisticated attacks.



Steps Involved in E-Skimming Attacks

The **Recorded Future Annual Payment Fraud Intelligence Report: 2023**[1] also speaks to this trend, noting the key findings summarized in *Table 1*

Evolving Magecart Tactics: In 2023, Magecart groups, known for their web skimming attacks, enhanced their tactics, techniques, and procedures (TTPs) to better conceal e-skimmer infections, making detection even more challenging.
Targeting of US Merchants: The report identified that US merchants were the primary targets of these breaches, although merchants in other developed e-commerce markets also faced significant risks.
Advancing Fraud Techniques: Looking forward to 2024, fraudsters are expected to continue refining their methods, leveraging a combination of advanced technical solutions and social engineering to bypass traditional fraud detection systems.

Table 1: Key Findings from Recorded Future Annual Payment Fraud Intelligence Report: 2023

These findings underscore the critical need for the enhanced security controls introduced in PCI DSS 4.0, specifically requirements 6.4.3 and 11.6.1, to protect against the evolving threats in the payment card industry.

[1] https://go.recordedfuture.com/hubfs/reports/cta -2023-1221.pdf

Breakdown of new requirements

With an understanding of the rationale behind PCI DSS 4.0 requirements 6.4.3 and 11.6.1, let's now delve into a detailed breakdown. These requirements are divided into four key components, each critical for strengthening the security of your payment pages and effectively countering the sophisticated threats present in today's digital landscape.

6.4.3 (a) - Script Inventory

Understanding What Runs on Your Payment Page

Web skimming attacks often exploit compromised JavaScript, making it essential for organizations to have complete visibility into every script operating on their payment pages. The primary requirement of this robust defence is to maintain a detailed inventory of all JavaScripts loaded on these pages, coupled with a thorough understanding of the purpose and functionality of each script.

By diligently managing this inventory, you can identify and eliminate unnecessary scripts, and more importantly, detect any that may be performing unauthorized or suspicious actions—significantly reducing the risk of data breaches through unauthorized data capture and safeguarding your customers' data.

6.4.3 (b) - Script Authorization

Ensuring Only Trusted Scripts Operate

Once you've established a script inventory, it's crucial to implement a mechanism guaranteeing that only these authorized scripts are allowed to execute on your payment pages. You can achieve this through various approaches, but the key is having a control system in place.

This control is particularly paramount as many skimming attacks involve unauthorized scripts injected from attacker-controlled domains. These malicious scripts are designed to steal sensitive information, such as credit card details. By enforcing strict script authorization, you create an impenetrable barrier that prevents such unauthorized scripts from executing, thereby significantly reducing the risk of data breaches and protecting your customers' sensitive information.

6.4.3 (c) - Script Integrity

Monitoring Authorized Scripts for Malicious Behavior

Authorization is only the first line of defense—ensuring that authorized scripts maintain their integrity is equally critical. Even legitimate scripts can be tampered with to perform unauthorized actions, such as capturing and transmitting sensitive data. For instance, while Google Analytics might be allowed to track user behavior, it should never access or transmit credit card information.

There have been cases where legitimate scripts were compromised, altering their behavior to perform malicious activities. To counter such threats, it's essential to continuously monitor the behavior of all authorized scripts on your payment page. If any script deviates from its expected function, immediate action must be taken to neutralize the threat to protect customers' sensitive information.

11.6.1 - Page Integrity

Safeguarding Against Sophisticated Web Skimming

Web skimming attacks have evolved beyond merely capturing credit card data from payment fields. Modern attackers now employ more advanced tactics, such as fake form attacks, where a fraudulent payment form is presented to the user to steal their credit card information before the legitimate form even appears.

To combat these sophisticated threats, it's crucial to ensure the integrity of your entire payment page. Under PCI DSS 4.0.1, the Page Integrity (11.6.1) requirement requires a holistic approach, involving monitoring all resources loaded on the page, the legitimacy of displayed forms, and the verification of HTTP headers sent by the server. A robust system should continuously monitor these elements and provide prompt alerts to site owners. This way, they can swiftly detect and address unauthorized changes, ensuring that potential threats are identified and mitigated before they can skim customer card data from the payment page.

Expert Insight

Implementing the controls outlined in PCI DSS 4.0 requirements 6.4.3 and 11.6.1 is not just about compliance—it's a strategic move to bolster the security of your payment pages.

By understanding and applying these four key controls, you can significantly mitigate the risk of client-side attacks and safeguard sensitive customer data. As you integrate these requirements into your security framework, consider how they align with your broader security and privacy goals and contribute to continuous improvement in your security posture.

Technical Options for Compliance

To meet PCI DSS 4.0.1 client-side requirements and effectively secure your payment pages, several standard technical approaches can be employed. These include **Remote Scanning**, **Content Security Policy (CSP)**, and **JavaScript Agent**. Choosing the right approach, or a combination of approaches, depends on your organization's specific security and process needs. Here's an overview of the options:

Remote Scanning

Remote scanning involves simulating an end-user's journey to the payment page, using automation tools like *Puppeteer* or *Selenium*. These tools automatically drive the browser through the site and arrive at the payment page, while a specialized data collection system attached to the browser monitors and captures comprehensive details about every interaction within the payment page. This includes loaded scripts, iframes, images, CSS, fonts, input forms, and HTTP headers set by the server, etc.

When properly processed and utilized, the collected data can be instrumental in helping organizations meet PCI DSS 4.0.1 requirements 6.4.3 and 11.6.1.

Advantages

1. **Ease of Use:** It's the simplest and fastest approach, as it requires no installation, setup, or configuration on your website.

2. **No Performance Impact:** As a fully remote system, it imposes no performance or functional impact on the site.

3. **Cross-Origin Visibility:** It can monitor third-party or cross-origin iframes and their influence on the payment page.

4. **Comprehensive Monitoring:** It monitors behaviors caused by HTML, CSS, and JavaScript, covering everything from low-level events like CSS-loaded images to high-level events such as scripts performing keylogging or reading input fields.

5. **HTTP Header Analysis:** It is capable of inspecting HTTP headers sent by the server.

6. **Detailed Behavioral Insights:** It can potentially offer detailed observations of page behavior, which is extremely valuable for thorough investigation and analysis by your security team.

7. **Data Privacy:** The system uses a test account to access the site, avoiding interactions with actual customer data.

Disadvantages

1. **Limited Session Coverage:** Remote monitoring doesn't capture every user session, which means specific behaviors triggered by certain conditions may go undetected. For example:

a. A script triggered only for users in New York won't be detected unless the scan is conducted from that location.

b. Special functionality reserved for high-spending customers (e.g., those who have spent over $5,000) won't be analyzed unless the test account meets this criterion.

2. **Lack of Blocking Capability:** The system can report unauthorized or malicious behavior, but cannot block it in real-time.

3. **Challenges with Anti-Automation Measures:** Sites with anti-automation measures can complicate the monitoring process.

4. **Need for Updates:** The scanning script requires regular updates to stay aligned with changes in user journeys or site navigation.

Expert Insight

To ensure your remote monitoring system serves you effectively and delivers optimal results, site owners should:

- **Verify Comprehensive Data Capture**: Confirm that the system accurately captures all relevant details—such as iframes, resources, input fields, and HTTP headers—by reviewing scan results.
- **Ensure Automatic Adaptation**: The system should automatically adjust to any changes in site design or

user journeys, ensuring monitoring remains uninterrupted and effective.

Content Security Policy (CSP)

Content Security Policy (CSP) acts as a firewall for your web pages, built into all modern browsers as a web standard. It allows you to specify a whitelist of authorized third-party domains that your payment page can safely interact with, ensuring that only trusted sources are permitted to execute on your site. CSP helps detect or block any unauthorized third-party interactions that fall outside of this list, preventing access to sensitive data by unapproved entities.

Advantages

1. **Higher Security Assurance**: CSP is built into modern browsers, making it less likely to be bypassed compared to proprietary solutions.
2. **Real-Time Monitoring and Protection**: CSP offers real-time oversight, ensuring that all user sessions on the payment page are monitored and protected from unauthorized interactions.
3. **Minimal Performance Impact**: As a native browser feature, CSP imposes virtually no performance impact on the site, ensuring smooth functionality.

4. **Applies to All Resources**: CSP applies to all resources, whether declared in the original HTML or dynamically loaded via JavaScript, ensuring comprehensive protection across your web page.

5. **No Functional Impact in Monitoring Mode**: When CSP is deployed in monitoring mode, it does not interfere with the functionality of the website, allowing for secure monitoring without operational disruption.

6. **Whitelist-Based Filtering**: CSP enforces a whitelist-based filtering system, providing a secure method to control which external resources can interact with your site.

7. **Effective Cross-Origin Redirect Protection**: CSP prevents unauthorized cross-origin redirects. For example, if a script allowed from a.com is redirected to b.com (not authorized by CSP), this violation will trigger an alert, and appropriate actions will be taken.

8. **Same-Origin iframe Coverage**: CSP extends protection to same-origin iframes, such as srcdoc iframes, enhancing the security of embedded content.

9. **Handles Violations During Page Unload**: CSP continues to monitor and enforce policies even when a page is unloading, ensuring full protection until the session ends.

Disadvantages

1. **Challenging for Dynamic Sites**: CSP's strict whitelist-based filtering can be difficult to maintain on dynamic websites. If the blocking mode policy is outdated, it may break site functionality by preventing new, legitimate resources from loading.

2. **Challenges with Single Page Applications (SPA):** Deploying CSP on SPAs, especially when applied only to specific PCI DSS 4.0.1 scope pages, can present challenges due to client-side navigation.

3. **Sparse Violation Information**: When a behavior violates a CSP policy, the system provides limited information, making it challenging to identify the root cause and investigate the issue further.

4. **Limited Behavioral Monitoring:** CSP cannot monitor or control high-level behaviors such as keyloggers or scripts that read sensitive input data, like credit card information.

Expert Insight

To ensure your Content Security Policy (CSP) serves you effectively and delivers optimal results, site owners should:

- **Simplify Deployment**: Consider deploying CSP only on sections of the site under PCI DSS 4.0.1 scope, such as the checkout page, rather than applying it across the entire website.
- **Engage Expert Resources**: CSP policy creation and management is nuanced; it's advisable to engage internal or external experts to optimize the policy.
- **Implement Practical Policies**: If a full blocking mode CSP policy isn't feasible, use a monitoring mode CSP policy or combine it with a limited blocking mode policy to balance protection and site functionality.
- **Leverage CSP Monitoring**: Maximize the benefits of CSP by utilizing a CSP report monitoring service tailored specifically for PCI compliance.

JavaScript Agent

A JavaScript Agent is a specialized piece of code that, when loaded onto a website, injects itself into critical JavaScript APIs on the page. This enables the agent to monitor and, if necessary, control the actions of other scripts running on the same page, providing an additional layer of security.

Advantages

1. **Real-Time Monitoring:** JavaScript Agents provide continuous, real-time monitoring and protection for every user session where it is loaded.

2. **High-Level Behavior Monitoring:** The JavaScript Agent can monitor the actions of other scripts in real-time, enabling it to detect and block high-level behaviors such as keylogging or unauthorized access to input fields, including credit card data.

3. **Detailed Behavioral Reporting:** The system offers detailed insights into which script is responsible for specific behaviors, making it easier to identify and address potential threats.

4. **Flexible Filtering:** The system has the potential to support both blacklist and whitelist filtering, which would simplify management and minimize the risk of disrupting site functionality.

5. **Support for Single Page Applications (SPA):** JavaScript Agents can recognize client-side navigation in SPAs and apply protections only to the relevant sections of the page, such as payment fields within the PCI DSS scope.

Disadvantages

1. **Proprietary Solution**: JavaScript Agents are not a web standard, meaning each vendor has their own implementation and features. This makes it more challenging to evaluate and compare solutions effectively.

2. **Limited to JavaScript Layer**: Since JavaScript Agents operate within the JavaScript layer, they cannot monitor behaviours outside of this scope. For example, they cannot track scripts embedded directly in the original HTML sent by the server or resources like images or fonts loaded via CSS.

3. **Monitoring Limitations Due to Load Order**: The JavaScript Agent can only monitor scripts that load after it. Any scripts that load before the agent are not monitored and can bypass the agent's restrictions.

4. **Potential for Bypass in Complex Scenarios**: Even within the JavaScript layer, certain complex behaviors may bypass the agent's monitoring. For instance, if a large amount of text is dynamically added to the innerHTML property of an element, the agent may struggle to parse all the content without causing significant performance issues. Attackers could exploit this limitation to evade detection.

5. **Limitations During Page Unload**: When a page is being unloaded, the browser imposes restrictions that prevent the JavaScript Agent from reliably communicating or capturing behaviors that occur during this time.

6. **Possible Session Interference from Header Monitoring**: To monitor HTTP headers, as required by PCI DSS 11.6.1, JavaScript Agents issue a fetch request to the server to inspect the response headers. Depending on the site's design, this can interfere with the user session's state, potentially breaking site functionality.

7. **Performance Impact**: Deploying a JavaScript Agent can introduce a non-trivial performance impact, which may be problematic for websites that have a low tolerance for performance degradation.

8. **Data Verification Challenges**: Because JavaScript Agents are proprietary solutions, verifying the nature and security of the customer data they collect is more complex compared to standard solutions.

Expert Insight

To ensure your JavaScript Agent serves you effectively and delivers optimal results, site owners should:

- **Optimize Placement for Maximum Coverage**: Load the JavaScript Agent at the top of the page to ensure it monitors all subsequent scripts for maximum protection.
- **Review Data Collected**: Regularly review the data collected by the JavaScript Agent and transmitted to external servers, ensuring it aligns with your data security and privacy expectations.
- **Measure Performance Impact**: Assess the performance impact of the agent once it's fully configured on your site, ensuring it operates within your acceptable performance limits.
- **Evaluate Alert Quality**: Ensure that the alerts generated by the system provide clear, actionable insights for thorough investigation and timely action.

Comparative Analysis: CSP vs. JavaScript Agent

Staying compliant with PCI DSS 4.0.1 requirements, particularly 6.4.3 and 11.6.1, requires the right technical approach. With growing attention on real-time protection options, choosing the right solution—or a combination of solutions—depends on your organization's specific needs. This comparative analysis highlights the key differences between Content Security Policy (CSP) and JavaScript Agents, offering insights to help you determine the best approach for meeting both compliance and security objectives.

1. Impact on Performance

CSP	JavaScript Agent
As a native browser feature, CSP imposes virtually no performance impact on the site.	JavaScript Agents can introduce a non-trivial performance impact, which may be problematic for websites with low tolerance for performance degradation.
Winner: **CSP** – Near-zero performance impact ensures smoother site functionality.	

2. Real-Time Monitoring and Protection

CSP	JavaScript Agent
CSP offers real-time monitoring, ensuring that all user sessions on the page are monitored and protected from unauthorized interactions.	JavaScript Agents provide continuous, real-time monitoring for every user session where they are deployed, offering in-depth visibility into script actions and potential threats.
Winner: **Draw** – **Both** solutions offer real-time monitoring, but JavaScript Agents provide more detailed insights into script behaviors.	

---

3. Security Assurance

CSP	JavaScript Agent
Built into modern browsers, CSP has a higher assurance level and is less likely to be bypassed compared to proprietary solutions.	JavaScript Agents, as proprietary solutions, are more susceptible to being bypassed due to varied implementations.
Winner: **CSP** – A native browser feature, CSP provides a stronger security foundation.	

---

4. Behavioral Monitoring

CSP	JavaScript Agent
CSP cannot monitor or control high-level behaviors, such as keylogging or scripts reading sensitive input data.	JavaScript Agents can detect and block high-level behaviors like keylogging or unauthorized access to input fields, including credit card data.
Winner: **JavaScript Agent** – Its ability to monitor high-level behaviors makes it more comprehensive for detecting sophisticated threats.	

---

5. Coverage of Resources

CSP	JavaScript Agent
CSP applies to all resources, whether declared in the original HTML or dynamically loaded via JavaScript.	JavaScript Agents are limited to monitoring scripts within the JavaScript layer and cannot track resources loaded via CSS or embedded directly in HTML.
Winner: **CSP** – Broader resource coverage ensures protection across all web components.	

---

6. Handling Violations During Page Unload

CSP	JavaScript Agent
CSP continues to monitor and enforce policies even during the page unload process.	JavaScript Agents are restricted by browser limitations during the page unload process, preventing them from reliably capturing or communicating behaviors.
Winner: **CSP** – Continuous protection during the page unload provides uninterrupted security.	

---

7. Cross-Origin Redirect Protection

CSP	JavaScript Agent
CSP prevents unauthorized cross-origin redirects. For example, if a script allowed from a.com redirects to b.com, which is not authorized, CSP triggers a violation alert.	JavaScript Agents do not provide cross-origin redirect monitoring.
Winner: **CSP** – Stronger cross-origin redirect protection ensures safer site interactions.	

---

8. Whitelist/Blacklist Filtering

CSP	JavaScript Agent
CSP enforces a strict whitelist-only filtering system, allowing only pre-approved external resources to interact with your site. This approach is secure but can be restrictive, making it challenging to quickly accommodate legitimate new resources.	JavaScript Agents have the potential to support both whitelist and blacklist filtering, offering greater flexibility in managing which scripts or resources are allowed or blocked. This flexibility reduces the risk of disruptions to site functionality and eases operational management.
Winner: **JavaScript Agent** – Greater flexibility with both whitelist and blacklist filtering allows for more effective control over resources.	

9. Violation Reporting and Behavioral Insights

CSP	JavaScript Agent
CSP provides limited information when a behavior violates its policy, making it challenging to identify the root cause and investigate further.	JavaScript Agents provide detailed insights into which script is responsible for specific behaviors, aiding in faster identification and resolution of potential threats.
Winner: **JavaScript Agent** – Detailed behavioral reporting offers actionable insights for quick remediation.	

10. Data Transparency and Verification

CSP	JavaScript Agent
With CSP, the nature of data collected and shared is well-documented and standardized.	JavaScript Agents, being proprietary solutions, make it harder to verify and evaluate the data they collect, adding complexity to ensuring data security.
Winner: **CSP** – As a web standard, CSP ensures greater transparency and data verification.	

11. Flexibility in Dynamic Environments

CSP	JavaScript Agent
CSP's rigid whitelist-based filtering can be challenging to maintain on dynamic websites that frequently change scripts or content. If the whitelist is not updated, legitimate resources may be blocked, causing functionality issues.	JavaScript Agents offer more flexibility with the potential for both whitelist and blacklist filtering, as well as the ability to apply these filters to high-level behaviors like keylogging, making them more adaptable to dynamic sites.
Winner: **JavaScript Agent** – The combination of filtering flexibility and the ability to apply filters to high-level behaviors makes it better suited for dynamic environments.	

12. Support for Single Page Applications (SPA)

CSP	JavaScript Agent
Deploying CSP on SPAs can be challenging, especially when applied only to specific sections of the site within the PCI DSS scope. Client-side navigation can complicate the enforcement of CSP policies.	JavaScript Agents can dynamically recognize client-side navigation in SPAs and apply protections only to the relevant sections of the site, such as payment fields within the PCI DSS scope.
Winner: **JavaScript Agent** – Its ability to handle client-side navigation makes it more suitable for SPAs, ensuring precise application of protections.	

Secure Your Payment Pages and comply with PCI DSS 4.0.1

The information we've covered should provide you with a solid understanding of the new PCI DSS 4.0.1 controls—specifically Sections 6.4.3 and 11.6.1—why they were introduced, the challenges they address, and the various technical approaches available to meet these requirements. Armed with this knowledge, you are now in a stronger position to make informed decisions about the best solutions to secure your payment pages.

Now that you're equipped to make an informed decision, we invite you to explore Domdog and its capabilities for meeting this requirement.

How Domdog Can Simplify Your Compliance Journey

Domdog is a dedicated page security provider offering the most flexible solution to meet your organization's PCI DSS 4.0.1 compliance needs, specifically for requirements 6.4.3 and 11.6.1. Unlike traditional solutions that require you to choose between different approaches—often leading to compromises—Domdog integrates all 3 compliance approaches outlined earlier, ensuring seamless alignment with your unique preferences, internal processes, and operational constraints. Our solution is designed to flexibly adapt to your unique preferences, internal processes, and practical constraints.

With Domdog, you can navigate compliance through a tiered approach, offering 5 distinct grades of compliance. You can start with Grade 1 compliance, which can be implemented instantly, and progressively enhance your security measures up to Grade 5—the most robust security level. Whether you're just beginning your compliance journey or looking to elevate your security, Domdog provides you with a flexible solution that evolves with your needs.

To help you get started, we're offering Grade 1 compliance completely free until March 31, 2025—no strings attached.

Domdog's Flexible Compliance Options

Domdog simplifies your audit process by providing a comprehensive single-page report, including all necessary information and supporting evidence to demonstrate compliance with PCI DSS 4.0.1 requirements 6.4.3 & 11.6.1.

Grade 1	Remote Monitoring Only
Grade 2	Partial Real-time Monitoring + Remote Monitoring (*CSP or JavaScript Agent*)
Grade 3	Full Real-time Monitoring + Remote Monitoring (*CSP and JavaScript Agent*)
Grade 4	Partial Real-time Protection + Remote Monitoring (*CSP or JavaScript Agent*)
Grade 5	Full Real-time Protection + Remote Monitoring (*CSP and JavaScript Agent*)

Grade 1: Effortless Compliance

Remote Monitoring Only

Domdog will remotely scan the sections of your site that fall under PCI DSS 4.0.1 compliance scope, requiring no setup,

installation, or configuration on your end. Scanning can begin instantly, and within minutes, you'll have access to our dashboard, allowing you to achieve compliance with just a few clicks.

Grade 2

Partial Real-time Monitoring *(CSP or JavaScript Agent)* + Remote Monitoring

Building on the foundation of remote scanning, Grade 2 adds real-time monitoring for enhanced coverage of user sessions. Deploy Domdog's JavaScript Agent or implement a monitoring mode CSP, using either our policy or your existing one. . This step-up significantly improves compliance quality by extending monitoring to every user session.

Grade 3

Full Real-time Monitoring *(CSP and JavaScript Agent)* + Remote Monitoring

Grade 3 offers a robust combination of remote monitoring alongside full real-time monitoring through both JavaScript Agent and CSP in monitoring mode. Utilize your existing CSP policy or the one provided by Domdog. This dual approach delivers unparalleled insight into every user session, significantly strengthening your compliance posture without compromising site functionality.

Grade 4

Partial Real-time Protection *(CSP or JavaScript Agent)* + Remote Monitoring

At Grade 4, Domdog introduces blocking mode through either CSP or JavaScript Agent, in addition to remote monitoring. You can leverage an existing blocking mode CSP or opt for a custom-tailored policy designed to minimize disruptions to site functionality. Our team will collaborate closely with you to develop the optimal solution that aligns with your requirements and operational constraints.

Grade 5: Ultimate Compliance

Full Real-time Protection *(CSP and JavaScript Agent)* + Remote Monitoring

Grade 5 represents the pinnacle of security, combining remote monitoring with both a blocking mode CSP policy and JavaScript Agent. This comprehensive approach offers the highest level of protection, integrating the best features of CSP, JavaScript Agent, and remote monitoring. If desired, you can gradually progress to this ultimate level of compliance over time, ensuring your payment page is fully secured against the most advanced threats.

Case Studies

E-Commerce Company Overcomes CSP Challenges

Challenge: An e-commerce company struggled with their existing CSP implementation.

Solution: Domdog's strategy involved implementing a CSP in monitoring mode, providing visibility without compromising functionality. Gradually, we introduced a combination of monitoring mode and minimal blocking, allowing the company to maintain functionality while improving security. Over the course of a year, the company transitioned fully to blocking mode CSP, achieving robust, compliant security without sacrificing performance.

Global Fintech Company Navigates Compliance Without CSP or JavaScript Agent

Challenge: A major global fintech player needed to comply with PCI DSS 4.0 but had stringent performance requirements that precluded the use of JavaScript agents. Additionally, they already had a sophisticated in-house CSP solution and required detailed insights into the scripts on critical pages.

Solution: Domdog provided a remote scanning solution that delivered exceptional visibility into the scripts loaded on their

sites. This detailed reporting allowed the fintech company to track script behavior, identify responsible internal teams, and ensure no malicious scripts were present, all without compromising site performance.

Single Page Application (SPA) Compliance Without CSP

Challenge: A client with a Single Page Application (SPA) faced difficulties using CSP due to its application across both critical and non-critical sections of the site, causing functionality issues.

Solution: Domdog offered a JavaScript Agent specifically designed to support SPAs. This solution helped achieve compliance by selectively monitoring and controlling script behavior on critical pages, bypassing the challenges associated with CSP.

In-House Solution Enhanced with Domdog's Expertise

Challenge: A platform servicing multiple merchants opted to build its own internal compliance solution but faced challenges in key areas.

Solution: Domdog collaborated with the platform by providing access to our web intelligence database and programmatic scanner. This empowered the platform to overcome their challenges and successfully complete their custom solution.

Managing Compliance for Hundreds of Payment Pages

Challenge: A large enterprise with hundreds of payment pages required a streamlined approach to bulk justification and management.

Solution: Domdog delivered a scalable solution that enabled efficient bulk management and justification of compliance across all payment pages. This ensured consistent security standards and simplified the auditing process.

Overcoming Engineering Resistance to JavaScript Agent Deployment

Challenge: A large e-commerce provider faced resistance from their engineering team regarding the deployment of a JavaScript Agent, despite security team approval. Concerns included a lack of transparency, potential site behavior impacts, and risks associated with using a third-party solution.

Solution: Domdog addressed these concerns by showcasing the JavaScript Agent's key features: high levels of control, a lightweight design, the ability to host the agent on the customer's own servers, and control over updates. These reassurances helped resolve the engineering team's reservations, leading to the successful deployment of Domdog's JavaScript Agent.

Domdog's Implementation of All 3 Approaches

This section outlines how Domdog developed each of these three approaches and highlights the unique advantages they offer compared to other solutions in the industry.

Remote Scanner	CSP	JavaScript Agent

Domdog's Remote Scanning Approach

Domdog leverages a specialized browser running on our cloud servers to execute remote scanning. This browser is fully automated, systematically collecting detailed information on every page it visits. It replicates the entire user journey across your site, ensuring that every page within the PCI DSS 4.0.1 scope is thoroughly and accurately monitored.

Key Highlights

Comprehensive Behavioral Reporting: Domdog's scanner doesn't just scan behaviors; it provides exhaustive details for each action. For instance, if a site initiates a fetch request to an external domain, you'll see the actual HTTP request, along with all the data sent.

Screen Recording of Events: We provide a screen recording of the entire browsing session, highlighting exactly when and how specific events occurred, giving you a visual timeline of the user journey.

Precise Code Identification: For each event detected, Domdog pinpoint the exact line of code in the script responsible for the action. Site owners can review the script's code and behavior history, enabling deeper analysis and understanding of potential issues.

Key Benefits

When a new behavior is reported, analysts must investigate and verify its legitimacy. This process can be time-consuming and complex without adequate information.

Domdog's remote scanning approach aims to streamline this process by providing detailed, actionable reports for a thorough investigation. These reports offer more comprehensive data than what is typically available through browser developer tools, significantly reducing the time and effort required to assess potential threats.

Domdog's CSP Approach

Domdog offers a specialized CSP Report Monitoring service, purpose-built to detect web-skimming attacks and ensure full compliance with PCI DSS 4.0.1 requiremnts 6.4.3 and 11.6.1. Whether you already have a CSP policy or need assistance in creating one, Domdog works closely with you to develop an optimal CSP policy tailored to your unique requirements.

Key Highlights

Unlimited CSP Reports: Unlike most CSP monitoring services that charge based on the number of reports processed, Domdog offers unlimited CSP reports, giving you comprehensive coverage without added costs.

Report Aggregation: Domdog takes raw CSP report data and transforms it into actionable insights that are crucial for attack detection and PCI compliance.

Threat Intelligence-Based Alerts: Domdog uses a threat intelligence database to assess new behaviors identified through CSP reports, automatically generating alerts that correspond to the threat level.

Support for Multiple Policies and Versions: Domdog supports sites with multiple CSP policies, allowing you to group reports by Policy ID and Version. This feature simplifies the review process and aids in making necessary updates.

Policy Integrity Monitoring: Domdog tracks the integrity of your CSP policies, sending alerts if a policy is removed or modified. This ensures that your site remains protected at all times, with no gaps in security.

Key Benefits

CSP implementation can pose challenges, such as the risk of breaking site functionality, managing policies over time, and interpreting violations.

Domdog's approach addresses these concerns by combining expert assistance with a specially designed CSP monitoring system. This ensures that your site remains secure without sacrificing usability, while also simplifying the management and analysis of CSP policies.

Domdog's JavaScript Agent Approach

Domdog's JavaScript Agent is designed to monitor and control the behavior of other scripts on your site. Given the proliferation of third-party scripts, configuring and managing these can become a complex and time-consuming task for site owners. Domdog simplifies this process by focusing on protecting critical assets on your payment page. Instead of managing the behavior of every individual script, Domdog identifies and safeguards essential elements—like cardholder data—ensuring that only authorized scripts can interact with these assets. This targeted approach makes configuration and management much more straightforward and sustainable over time.

Key Highlights

Granular Control: Domdog's JavaScript Agent offers granular control over its functionality, allowing for precise management of script behavior.

Blacklist and Whitelist Support: Domdog's agent supports both blacklist and whitelist filtering, enabling flexible and easy management of allowed and disallowed behaviors.

Transparency in Data Collection: Domdog ensures full transparency into the data collected by the JavaScript Agent, giving you complete oversight and control over what's being monitored.

Flexible Hosting Options: The agent can be served from Domdog's CDN or hosted and served from your own CDN, providing flexibility based on your preferences.

Key Benefits

Using a JavaScript Agent often raises concerns about performance impact, transparency, and data collection. Domdog's JavaScript Agent is specifically engineered to minimize these concerns. By offering granular control, full transparency, and flexible hosting options, Domdog ensures that your site's performance remains unaffected while providing the robust security necessary to protect your critical assets.

About Domdog

Page Security & Privacy Platform

Our journey in **webpage security** began with the realization that modern web functionality, while transformative, comes with inherent security trade-offs. Over the past decade, we've been at the forefront of addressing these challenges, leveraging cutting-edge research and engineering to deliver robust, client-side security solutions.

Now as a **Page Security & Privacy Platform**, Domdog continues to protect billions of user sessions. Our **three modes of monitoring** provide comprehensive, research-driven solutions tailored to combat modern web page threats.

For more information:

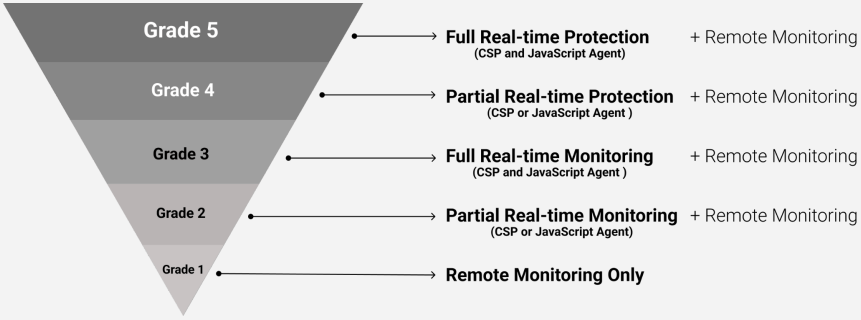
- ☐ **Website:** https://domdog.io
- ☐ **Email:** sukesh@domdog.io

Built by engineers, for engineers. At Domdog, we believe that effective security should be both simple and transparent.

DOMDOG

Do you need to comply with PCI DSS 4.0.1 requirements 6.4.3 and 11.6.1?

We can help.



To help you get started, we're offering Grade 1 compliance completely free until March 31, 2025—no strings attached.

CSP or Remote Scanning or Script-based Solution!

Trying to decide what's best for your organization?

Get all three in a single product with Domdog, the most flexible and no-compromise solution for 6.4.3 & 11.6.1 compliance.

Enjoy Free Compliance till March 2025!

To learn more:

Contact us

DOMDOG
Page Security & Privacy Platform

- **100 Billion+** CSP Reports Processed
- **20 Billion+** Page Views Protected
- **15 Years of** Webpage Security Expertise

https://domdog.io

sukesh@domdog.io