

Why we're different

What criteria should I use to evaluate a client-side security solution for PCI compliance?

When you compare security solutions you need to look beyond headline promises and examine how each product capabilities performs against how modern day client-side attacks work and to the operational realities of the business.

(1) Ask how it handles attacks as they really happen. Can it spot new threats the moment they land, understand the payload of the script, and follow its behavior as it changes with the user or the time of day?

(2) Can they show exactly what each third-party script collects and still detect a malicious payload that fires for only 1 in 1,000 visitors, or just 5% of users after 5 p.m.?

(3) Does it remember every action for forensics later, guard live sessions where customers type passwords and card numbers?

































(4) Will it catch sneaky DOM tricks, watch exactly the code your visitors run, and use AI to deobfuscate code in real time or will they just use threat feeds where malicious JavaScript can stay undetected for months and years?

If the answer to any of these is no, your security team will be left guessing, let alone prevent a client-side attack from happening. Looking at each of these capabilities up front is the quickest way to know whether a product will protect your customers and your bottom line.

We know that there is a lot of marketing content out there, but the proof is in the pudding. You should always write a malicious script yourself to see whether the solution catches it. If you need us to, we can share one with you too. We'd like for you to use a solution that actually works.

In the scope of PCI, some solutions may have some of the required data scattered around their dashboard. To prevent you from having to keep track of script justifications in a spreadsheet you may want to consider the UI in relation to the PCI requirements.

Want practical implementation advice? [Talk to an expert](#)

Criteria	Why It Matters	What the Consequences Are	CSP	Crawler	JS-Based	Hybrid Proxy
Real-time Protection	Attacks can occur between scans or in the excluded data when sampled	Delayed detection = active data breaches				
Full Payload Analysis	Ensures deep visibility into malicious behaviors within script code itself	Threats go unnoticed unless the source is known on a threat feed				
Dynamic Threat Detection	Identifies attacks that change based on user, time, or location	Missed detection of targeted attacks				
100% Historical Tracking & Forensics	Needed for incident response, auditing, and compliance	No way to prove or investigate past (missed) attacks and no ability to scale the scope of an incident to specific users				
No Performance Impact	Avoids trade-offs between performance and security	Higher page load times can reduce conversions and hurt UX				
Bypass Protection	Stops attackers from circumventing controls via DOM obfuscation or evasion	Stealthy threats continue undetected				
Certainty the Script Seen by User is Monitored	Aligns analysis with what actually executes in the browser	Gaps between what's reviewed and what's actually executed				
AI-driven Script Analysis	Detects novel or evolving threats through behavior modeling	Reliance on manual updates, threat feeds or rules = slow and error-prone detection				
Implementation Complexity & Timeline	Impacts time-to-value and internal resource costs	Long deployment timelines reduce agility	High	Medium	Medium	Low
Can meet 11.6.1 requirement	11.6.1 relates to monitoring changes in the security headers as well as the script contents themselves	Not monitoring security headers violates 11.6.1—missing or altered headers signal potential attacks.	