

HIDDEN POINTS OF PUBLIC PRESENCE

Your homepage is not your public attack surface. It is merely the public endpoint everyone remembered to name.

PRACTITIONER POSITION

A point of public presence is any Internet-reachable address, name, service, route, protocol, or provider-mediated entryway through which an external party or service can reach an entity-operated or entity-controlled capability. Some are obvious websites. Others are webhook receivers, callback URLs, API stages, CDN or load-balancer origins, serverless function URLs, identity endpoints, file-transfer services, remote-access gateways, alternate ports, recovery sites, legacy names, vendor-configured targets, or nonproduction systems that quietly became reachable. PCI SSC's current ASV Program Guide requires scan customers to supply all external-facing IP addresses, fully qualified domain names, and other unique entryways into applications for the entire in-scope infrastructure, including hidden directories that ordinary crawling may not find. PCI DSS penetration-testing scope and application-security obligations answer different questions and require their own fact-specific evaluation. My practitioner position is that discovery should begin with the broad population of public presences having a confirmed or plausible relationship to the CDE or its security, then preserve separate dispositions for PCI scope, ASV or other external scanning, penetration testing, software inventory, secure development, change management, monitoring, provider coverage, and removal. Public reachability is a discovery fact-not an automatic compliance conclusion. But an endpoint omitted before analysis never received a defensible conclusion at all.

THE FIELD PATTERN

WHAT THIS LOOKS LIKE IN PRACTICE

The payment program submits the same four IP addresses for its quarterly external scan. A certificate inventory reveals six additional names. One points to a disaster-recovery load balancer. Another reaches a serverless callback that receives payment status events. A third is a CDN origin that bypasses the public WAF when addressed directly. The telephony provider has an HTTPS action URL stored only in its console. A retired API hostname still resolves, an IPv6 listener exists beside an IPv4-only target list, and a support engineer temporarily exposed a diagnostic route during an incident. None appears in the homepage crawl. Several may ultimately be outside PCI DSS scope; several may be separately covered or subject to different tests. The immediate failure is simpler: the organization cannot show that anyone discovered, related, and dispositioned them.

SIGNALS DETECTED

WHAT SHOULD MAKE YOU STOP

- The public inventory begins and ends with production websites and their current IPv4 addresses.

- Webhook, action, status-callback, return, notification, and event-receiver URLs live only in provider consoles or environment variables.
- CDN, WAF, reverse-proxy, and load-balancer names are recorded, but direct origins and alternate routes are not.
- API gateway stages, serverless function URLs, managed-container endpoints, and default cloud hostnames are absent from the software and infrastructure inventories.
- Disaster-recovery, blue-green, canary, test, support, and legacy endpoints are called inactive even though the Internet can still reach them.
- The external target list covers IPv4 while DNS, cloud services, or network listeners expose IPv6.
- An endpoint is authenticated, allowlisted, obscure, or unlinked from the homepage and is therefore described as not public.
- The ASV list, broader external vulnerability scan, penetration-test scope, application inventory, and change-management population are maintained independently.
- Provider-hosted or provider-mediated endpoints disappear behind the provider name instead of being mapped to the exact service and responsibility model.
- A newly discovered public presence is closed in one scanner without reopening inventory, diagrams, ownership, testing, and scope decisions.

WHY IT MATTERS

THE COST OF LEAVING IT UNEXPLAINED

Start by separating the objects. A URL path is not a hostname. A hostname is not an IP address. An IP address is not the CDN, WAF, load balancer, origin, serverless function, container, application, or code that responds. One public journey can therefore produce several related inventory and testing objects. Flattening them into one row creates false coverage claims.

Use multiple discovery lenses. Reconcile current and historical DNS, certificate inventories, cloud and network interfaces, CDN and WAF configurations, load balancers, API gateways, serverless and container inventories, firewall and NAT rules, public storage and file-transfer services, identity application registrations, infrastructure as code, application configuration, provider consoles, webhook settings, logs, prior scans, penetration-test findings, and incident records. No single source proves completeness.

Treat authentication and obscurity honestly. A service can be publicly reachable even when it requires a signature, token, client certificate, source allowlist, VPN credential, or unguessable path. Those controls materially affect exposure and testing, but they do not erase the endpoint from discovery or ownership. A route that is absent from site navigation can still be an application entryway.

Resolve the PCI relationship before choosing the compliance treatment. Determine whether the presence stores, processes, or transmits account data; is in the CDE; connects to the CDE; can impact the security of the CDE; redirects, embeds, or influences a payment journey; administers or protects a relevant component; or remains genuinely unrelated. Keep plausible but unresolved relationships visible until an accountable decision is supported.

Keep testing populations separate. The current ASV Program Guide describes IP addresses, FQDNs, and unique application entryways supplied for in-scope infrastructure. Network vulnerability scanning generally addresses hosts and exposed services. External penetration testing evaluates the exposed perimeter and application behavior against the entity's methodology. A hidden route may matter to one, several, or none of those activities depending on the facts; copying one target list into every activity is not a methodology.

Bring custom endpoints into application governance. If entity-developed code terminates a webhook, callback, API, upload, or administrative route that is in scope or security-impacting, identify the application, repository, runtime, owner, deployment, dependencies, authentication and signature controls, secrets, logging, failure behavior, and changes. The fact that a TPSP initiates the request does not make the receiver provider-operated.

Map provider mediation without outsourcing the conclusion. Record which party owns the domain, DNS, certificate, edge, origin, code, secrets, configuration, monitoring, vulnerability response, and testing evidence. A provider AOC or responsibility matrix can inform that map; it cannot describe the customer's unlisted callback target or prove how the customer implemented it.

Test negative and alternate paths. Direct-origin access, alternate host headers, old DNS, IPv6, unexpected methods, retry and replay behavior, unsigned requests, expired credentials, error handlers, verbose diagnostics, forgotten API versions, staging deployments, and emergency routes frequently expose a different control boundary than the happy path.

Make change produce reconciliation. A new certificate, DNS record, listener, function URL, provider callback, load balancer, public bucket, API stage, redirect, recovery activation, or firewall rule should challenge the public-presence register and every affected inventory, diagram, scan, test, monitoring, and scope record. Discovery after deployment should be treated as a control signal, not merely a scanner ticket.

Reduce the population deliberately. Remove abandoned endpoints, disable direct origins, expire temporary routes, consolidate aliases, close unused protocols, limit management exposure, and document separately covered provider surfaces. The strongest disposition is often W/? : the organization no longer needs the public presence, so it eliminates it rather than writing a heroic explanation.

QUESTIONS

QUESTIONS THAT EXPOSE THE WACK

- 01 Which external party, service, or network can reach this point, by which name, address, protocol, port, path, and authentication condition?
- 02 What edge, origin, application, runtime, code, data, secret, and downstream service sit behind it?
- 03 Which independent discovery sources identified or failed to identify it, and what population boundary did each source actually cover?
- 04 What is its relationship to account data, the CDE, connected systems, security-impacting functions, payment redirection or embedding, administration, and protection?
- 05 Is it represented in the detailed hardware, software, cloud-resource, application, domain, certificate, and provider inventories under stable identifiers?

- 06 Which external vulnerability scan or ASV population evaluates its host, FQDN, address, service, or unique application entryway, if applicable?
- 07 Which external penetration-test proposition evaluates its actual application behavior, trust boundary, authentication, alternate path, and failure behavior, if applicable?
- 08 If custom code responds, which secure-development, dependency, change, vulnerability, logging, and deployment controls govern it?
- 09 What provider evidence or separate coverage applies, which customer responsibilities remain, and where is the implemented service configuration proved?
- 10 Should this point remain public at all, and which event must reopen every dependent scope and testing disposition?

THE CLUSTER OF WACK

FIND / MAP / EXPLAIN / REDUCE / PROVE

ESCAPE THE WACK

01 / FIND

Build candidate public presences from DNS, certificates, cloud inventories, network exposure, application configuration, provider consoles, code, infrastructure as code, logs, prior tests, and incident history. Include obscure, authenticated, nonproduction, alternate, and unresolved entryways.

02 / MAP

Separate URL, hostname, address, protocol, edge, origin, runtime, application, code, data, security influence, owner, provider, and environment. Join each point to inventories, diagrams, payment journeys, scans, tests, changes, and monitoring through stable identifiers.

03 / EXPLAIN

Record public reachability as an observation, then issue independent dispositions for PCI relationship, ASV and other external scanning, penetration testing, application and SDLC governance, provider coverage, and continued business need. Preserve evidence and the responsible authority for each decision.

04 / REDUCE

Eliminate abandoned and temporary exposure, close direct origins and unused protocols, retire legacy names, consolidate duplicate routes, restrict management surfaces, and correct provider or ownership gaps without deleting the history that explained the change.

05 / PROVE

Reconcile the current candidate population to inventories, diagrams, applicable scan submissions, penetration-test rules of engagement, application records, change history, provider evidence, and removal verification. Reperform discovery after meaningful change and disposition every difference.

THE FIELD LAW

THE LINE WORTH REMEMBERING

IF AN EXTERNAL SYSTEM CAN REACH IT BUT YOUR INVENTORY CANNOT, YOU HAVE FOUND A HIDDEN POINT OF PUBLIC PRESENCE.

Download the Hidden Public Presence Discovery and Disposition Register. Start with provider consoles and application configuration-not the homepage-and find the first callback, origin, alternate address, or recovery endpoint that no current testing population can explain.

TECHNICAL GROUNDING

START WITH THE SOURCE MATERIAL

These references ground the practitioner interpretation. They do not turn the metaphor or method into official PCI DSS terminology.

- WACK-SOURCE-0001 / PCI DSS v4.0.1 / PCI Security Standards Council / [Open source](#)
- WACK-SOURCE-0005 / PCI DSS Scoping and Segmentation Guidance for Modern Network Architectures / PCI Security Standards Council / [Open source](#)
- WACK-SOURCE-0006 / Resource Guide: Vulnerability Scans and Approved Scanning Vendors / PCI Security Standards Council / [Open source](#)
- WACK-SOURCE-0007 / PCI DSS Penetration Testing Guidance / PCI Security Standards Council / [Open source](#)
- WACK-SOURCE-0082 / Approved Scanning Vendors Program Guide v1.0 / PCI Security Standards Council / [Open source](#)

AUTHORITY CLASSIFICATION / PCI_SSC_SOURCE / ASSESSOR_JUDGMENT / PRACTITIONER_POSITION / PRACTITIONER_METHOD / AI_ASSISTED

Practitioner reviewed by Scott Norton on 2026-09-11 against canonical digest

504c18fdd83b01009e3475abf45dc83a4cc22097a89f041f61fd219d465e811d. Educational content, not a compliance determination.

Independent practitioner resource; not affiliated with or endorsed by PCI SSC. Authorship and authority classification do not determine legal intellectual-property ownership.