# Cloudflare Application Security

## Partner Services Bootcamp - Course Syllabus

---

### Course Overview

This five-day instructor-led bootcamp covers Cloudflare's application performance and security stack well beyond onboarding or certification prep. Everything is taught in chronological build order: each topic is presented in lecture, demonstrated live in the dashboard, then applied hands-on in a lab before the next topic begins. Days 1 through 4 work through the twelve modules and fifteen labs; Day 5 is a full-day individual capstone lab.

The labs follow the "AcmeCorp" scenario. The student is AcmeCorp's new Lead Application Security Engineer, inheriting an origin that sits on a public IP with no CDN, no real encryption, and no security layer. Over fifteen labs the student proves how exposed it is, routes and encrypts its traffic, makes it fast and resilient, builds a layered perimeter (WAF, bot management, rate limiting, API Shield, threat intelligence, and client-side protection), then seals the origin so the perimeter cannot be bypassed and replays the opening attacks to prove nothing gets through.

**Format:** Instructor-led with slides, live dashboard demos, lab exercises, and a full-day capstone lab **Duration:** 5 days (4 days instruction and labs, 1 full day capstone lab) **Schedule:** 9 AM to 4 PM daily, one-hour lunch **Lab environment:** Individual pre-onboarded Cloudflare Enterprise zones with a browser-based workstation for running test commands **Lab exercises:** ~55 hands-on exercises across 15 labs, plus the Day 5 capstone

---

### Who Should Attend

- Cloudflare partner engineers working with customers on application security deployments
- Solutions engineers who want deep, hands-on Cloudflare Application Services experience
- Network and security engineers responsible for WAF, bot management, rate limiting, and API protection
- Engineers migrating customers from Akamai (or other CDN / WAF vendors) to Cloudflare

## Prerequisites

- Familiarity with Cloudflare dashboard navigation
- Basic understanding of DNS and HTTP concepts
- Comfort reading command-line output (dig, nslookup, curl)

---

## Daily Schedule

Modules align to the current slide decks M00 through M11. Topics are taught in the order below.

| Day | Focus | Modules | Labs |
|---|---|---|---|
| **Day 1** | Foundations: Platform, DNS, and SSL | M00, M01, M02, M03 | Labs 1-3: The Exposed Origin, DNS Onboarding & Proxy Control, SSL/TLS & Encryption Standards |
| **Day 2** | CDN, Traffic Management, and Load Balancing | M04, M05, M06, M07 | Labs 4-6: Caching Foundations, Advanced Caching, Traffic Management & Rules Engine (Load Balancing is lecture + live demo) |
| **Day 3** | Security: WAF, Threat Intel, and Bots | M08, M09 | Labs 7-10: WAF Managed Rulesets, WAF Custom Rules, Bot Management, Threat Intel with Managed Lists |
| **Day 4** | Protecting APIs and Client-Side Security | M10, M11 | Labs 11-15: Rate Limiting, API Discovery & Endpoint Management, API Schema Validation, Page Shield, Seal & Validate the Perimeter |
| **Day 5** | Capstone Lab | -- | Full-day individual capstone: take AcmeCorp from exposed to sealed, checked by an automated grader |

---

## Module Descriptions

### M00: Introduction and Bootcamp Overview

Bootcamp logistics, learning principles, and the five-day schedule. The instructor establishes the teaching methodology: theory immediately followed by hands-on application, every lab modeled on a real-world customer scenario, and the see-one-do-one-teach-one approach. Cloudflare University is introduced as a resource for continued learning.

### M01: Application Performance

This module frames everything that follows. It covers how Cloudflare's global Anycast network operates, why every data center runs every service (the "single control plane" distinction from legacy CDN vendors), and the order of operations, the specific sequence in which a request passes through Cloudflare's modules. Understanding this flow is critical because it determines how DDoS mitigation, bot management, firewall rules, rate limiting, and API Shield interact with each other.

The module also covers compliance and data localization for regulated industries (GDPR, PII, the Anycast border-country edge case), multi-CDN architectures for financial institutions that require dual-CDN resiliency, and API automation fundamentals including Terraform, curl-to-Python workflows, and scoped API tokens with least-privilege permissions.

### M02: DNS

DNS resolution from root servers through authoritative name servers, with dig, nslookup, DNS Checker, and Dig Web Interface as troubleshooting tools. Full authoritative vs. partial (CNAME) DNS setups, the security tradeoffs of each (in a partial setup the root domain is often left exposed unless CNAME flattening or registrar-level redirect rules are used), CNAME flattening, and proxy status (orange cloud vs. gray cloud). Port limitations of the proxy for non-standard ports and the role of Cloudflare Spectrum for those cases.

### M03: SSL/TLS

The two certificates in every proxied request (edge and origin), all five SSL modes, and why **Full (Strict)** is the required best practice for production security. Universal, Advanced, and custom certificate types, auto-renewal, DCV records for pre-cutover activation, certificate precedence, and the conflict-resolution rule: if a customer uploads a custom certificate, keep the Universal certificate active as a backup so a forgotten custom-cert expiration does not become an outage.

### M04: CDN Part I

CDN fundamentals: pull vs. push CDN differences (relevant for Akamai migration conversations, since Cloudflare is a Pull CDN and pre-warming is generally unnecessary), how Cloudflare decides what to cache by default (based on file extension), verifying cache behavior with `cf-cache-status` headers, and why HTML isn't cached by default but often should be. Cache control directives, and the Cloudflare Trace tool for troubleshooting which rule matched a request.

### M05: CDN Part II

Cache keys and cache sharding, TTL strategy, cache-purge risks, and using Cache Performance analytics to find optimization opportunities. **Tiered Caching** (generic vs. Smart Tiered Caching) as an always-on best practice for shielding the origin. **Argo Smart Routing** for dynamic traffic acceleration, and

common mistakes like forgetting to enable Argo on Enterprise zones that purchased it (highly effective for global traffic, less useful for hyper-local traffic).

### M06: Traffic Management and Rules Engine

Creating and managing rules from the dashboard to move business logic off the origin and onto the edge: redirects, header manipulation, Host Header Override, geo-redirection, URL rewrites, and Managed Transforms for security response headers. Also introduces the "Last Match Wins" logic of Cache Rules (opposite of the top-down first-match Firewall Rules) that trips up new engineers.

### M07: Load Balancing, Health Checks and Steering

Pools, monitors, and steering policies (active-passive, dynamic Argo-based, geosteering), custom rules for path-based routing, session affinity, and content-aware health checks that catch "zombie" origins (a 200 response that is actually broken). Common pitfalls like overly aggressive health-check intervals that cause origins to flag Cloudflare IPs as a DDoS attack, and configuring expected codes to include 301/302 redirects (not just 200 OK).

### M08: Application Security

The largest module in the bootcamp, covering DDoS protection, WAF managed rulesets, the OWASP Core Ruleset, security analytics, custom WAF rules, and threat intelligence with managed lists.

- **DDoS Protection**: the automated always-on system that learns from ~20% of internet traffic, the difference between volumetric and signature-based detection, real customer scenarios where thresholds don't trigger (and how to handle that conversation), and the strong recommendation to leave sensitivity settings at defaults unless explicitly advised by support.
- **WAF Managed Rules**: the 728+ rule managed ruleset, why default mode is the best practice, the logging-first deployment workflow, zero-day protection (Log4j, Spring4Shell), and why exceptions are rarely needed.
- **WAF OWASP Ruleset**: anomaly scoring, the four paranoia levels (PL1-PL4), and the clear recommendation to leave OWASP off unless compliance (usually PCI or SOC 2) forces it on. If enabled, PL1 is safe for most sites; PL4 will block legitimate traffic (the "paranoid chihuahua" analogy).
- **Security Analytics**: reading the events dashboard, sampling limitations, filtering by action and Ray ID, and why log push to a SIEM is the true source of truth. The total-traffic dashboard for establishing normal baselines.
- **Custom Rules**: the match-and-action model, challenge types (interactive, JS, managed, with Managed Challenge preferred over Interactive Challenge for most situations), the "Log" action as the first step for any new rule, and the "Skip" action for bypassing security modules on trusted traffic. Firewall rules execute top-down first-match-wins, the opposite of Cache Rules.

- **Threat Intelligence with Managed Lists**: preemptively blocking anonymizers, open proxies, and botnet command-and-control using Cloudflare's continuously updated managed lists, with zero maintenance.

### M09: Bot Management

The core lesson: bot management determines "automated vs. human," not "good vs. bad." That makes it the wrong tool for API traffic (everything scores 1) and the right tool for applications expecting human interaction. Bot score (1 = definite bot to 99 = definite human), the three scoring-engine components (heuristics, machine learning, behavioral analytics), the ML model's evolution to today's version 8 with strong bimodal distribution, and verified bots.

The three-step implementation process is the operational heart of the module: (1) deploy a safe base rule in Log mode (Bot Score < 30, action Log), (2) collaboratively analyze traffic with the customer to identify known good bots (partners, internal QA tools) that score low and need exceptions, (3) move the main rule from Log to Block or Managed Challenge only after the traffic looks clean (anywhere from two days to five weeks). Also covers rule stacking, managed IP lists, challenge solve rates, and textbook attack patterns for credential stuffing and content scraping.

### M10: Protecting APIs: Rate Limiting and API Shield

Rate Limiting and API Shield as complementary tools for protecting API traffic that Bot Management can't help with.

- **Rate Limiting**: match criteria (IP, "IP with NAT support" to avoid blocking entire offices sharing a single public IP, origin response-code counting), threshold selection using the analysis tool or manual traffic review, the logging-first workflow, tiered rate-limit stacking, and distributed-counting tradeoffs.
- **API Shield**: capabilities in order of complexity: Discovery and Endpoint Management for visibility into API traffic (often revealing deprecated endpoints customers didn't know existed), Schema Validation as the quickest win (JSON/OpenAPI drops malformed requests immediately), fallthrough rules to fail closed on undefined operations, mTLS for mutual authentication, integrated Rate Limiting pre-populated from endpoint-management data, and Sequence Analysis / Sequence Mitigation for enforcing correct API call order using session identifiers (blocking thieves who skip directly from `/login` to `/transfer`). Implementation is a collaborative process with the customer's API developers, similar to bot management.

### M11: Page Shield: Client-Side Security

The last blind spot is the customer's own browser, where the WAF never sees the code that runs. Client-side and supply-chain attacks (a compromised third-party script or an injected skimmer) execute entirely in the browser. Page Shield gives visibility into every script, connection, and cookie loaded on

the page, surfaces malicious or unexpected resources, and lets the customer enforce an allowlist with a Content Security Policy (CSP) to close the supply-chain gap.

---

## Lab Exercises

Each student works in a dedicated pre-onboarded Cloudflare Enterprise zone with a shared, exposed origin serving the AcmeCorp website and a public orders API. A browser-based Ubuntu workstation (no SSH, nothing to install) provides the terminal for `curl`, `dig`, and `openssl` used to run attacks and verify each control. Because the origin is a shared machine the student cannot reconfigure, the few origin-side steps (installing an Origin CA certificate, failing over a pool, sealing the origin behind a firewall) are presented as guided concept walkthroughs; everything at the Cloudflare edge is hands-on.

The fifteen labs read top to bottom as one continuous story: exposed, onboarded, encrypted, fast, resilient, defended, sealed, and proven.

### Lab 1 - The Exposed Origin

*6 exercises*

Play the attacker to establish the "before" baseline. Hit the origin directly by IP, fingerprint its software, read a sensitive file it should never expose, scrape the product catalog, push a SQL injection string straight through, and knock on the locked admin panel. Nothing stops you. Every result is recorded so the same attacks can be replayed and defeated later.

### Lab 2 - DNS Onboarding & Proxy Control

*3 exercises*

Put Cloudflare in front of the origin: confirm authoritative DNS, proxy the web records so the real origin IP disappears, and learn when to leave a record unproxied (grey cloud) and why a grey cloud re-exposes the origin.

### Lab 3 - SSL/TLS & Encryption Standards

*3 exercises*

Fix encryption on both hops. Turn on edge encryption (Always Use HTTPS and a modern minimum TLS version), inspect the free Universal SSL certificate, move toward Full (Strict) with an Origin CA certificate, and order an advanced certificate for a deep subdomain Universal SSL cannot cover.

### Lab 4 - Caching Foundations

*4 exercises*

Take load off the struggling origin. Establish a caching baseline, offload a dynamic path with a cache rule, honor the developers' cache-control headers, and purge a stale cached response on demand.

### Lab 5 - Advanced Caching

*4 exercises*

Carry the site through a real surge. Cache entire HTML category pages, make that safe for logged-in users with a cookie bypass (and fix the most common cache-rule ordering trap), normalize marketing URLs with a custom cache key, then switch on Tiered Cache so the origin sees a fraction of the load.

### Lab 6 - Traffic Management & Rules Engine

*5 exercises*

Move business logic to the edge without touching origin code: localize visitors with a geo-redirect, stand up a campaign redirect, add a security response header, switch on Cloudflare's Managed Transforms, and rewrite a retired product URL to its replacement.

### Lab 7 - WAF Managed Rulesets

*5 exercises*

Start closing the Lab 1 holes. Prove two attacks still land through Cloudflare, deploy the managed rulesets the real-rollout way (Log first, then Block), add OWASP Core as defense in depth, discover the fix hard-blocks a legitimate customer, repair it with a surgical exception, and triage the whole incident in Security Analytics.

### Lab 8 - WAF Custom Rules

*6 exercises*

Enforce AcmeCorp's own policy as a layered set of custom rules: geofence sanctioned countries while quietly watching a competitor's region, challenge the abused contact form, lock down the admin panel the attacker reached in Lab 1, and let one trusted partner API through untouched, including the common "Allow action that doesn't actually block anyone" trap.

### Lab 9 - Bot Management

*3 exercises*

Stop the pricing scrapers that carry no header or country to block. Read the bot analytics, deploy a base challenge rule for low-scoring automated traffic, tune out trusted internal tooling, then predict and fix the blast radius of an over-broad bot rule.

### Lab 10 - Threat Intel with Managed Lists

*3 exercises*

Block sources that are known bad before they send a byte. Preemptively block anonymizers and open proxies with Cloudflare's continuously updated managed threat-intelligence lists, add a high-confidence block tier, and predict how it interacts with the trusted-partner bypass from Lab 8.

### Lab 11 - Rate Limiting

*2 exercises*

Handle attackers who pivot to volume. Cap requests to a hot page per IP to stop brute force, then throttle API-style fuzzing by counting the origin's own error responses.

### Lab 12 - API Discovery & Endpoint Management

*3 exercises*

Gain visibility you can't protect without. Configure a session identifier, generate traffic so API Shield learns the real API surface (including undocumented shadow endpoints), and bring those operations under management.

### Lab 13 - API Schema Validation

*2 exercises*

Stop trusting API callers. Onboard the orders API and enforce its OpenAPI contract at the edge so malformed requests are dropped before they reach the origin, and add a fallthrough block that rejects any operation the schema does not define.

### Lab 14 - Page Shield: Client-Side Protection

*3 exercises*

Close the browser blind spot. Turn on client-side resource monitoring to inventory every script and connection, spot a malicious script, then enforce a Content Security Policy allowlist to shut the supply-chain gap.

### Lab 15 - Seal & Validate the Perimeter

*3 exercises*

Prove the perimeter can't be bypassed. Seal the origin so it accepts traffic only from Cloudflare (simulated at the workstation on the shared lab origin), replay the Lab 1 attack suite, and watch each attack fail. Recap the architecture built across the course.

---

## Day 5: Capstone Lab

The course finishes with a full-day, individual, hands-on capstone lab (roughly six hours). Working solo in their own lab zone, each student takes a fresh, exposed AcmeCorp deployment from wide open to sealed: onboarding and hardening transport, building the layered WAF, bot, rate-limiting, API Shield, and client-side perimeter, then sealing the origin and validating that the opening attacks no longer get through. Submissions are checked by an automated grader that inspects both the configuration and the live behavior of the student's zone, so success is measured by a working, defensible deployment rather than a slide presentation. It is the practical exam for everything built across Labs 1-15.

---

## Optional Certification

Students are offered the opportunity to sit the Cloudflare Application Security Associate certification exam for free if they wish. The exam is not a required part of either bootcamp and neither course is structured as exam preparation; it is simply a no-cost chance to earn a Cloudflare certification. That said, several days of hands-on configuration, troubleshooting, and solution design give participants a strong practical foundation should they choose to take it.

---

## Migration Methodology (Interwoven)

Akamai to Cloudflare migration considerations are threaded throughout the security modules rather than taught as a single unit. The core philosophy is **"Move and Improve," not "Lift and Shift":** legacy Akamai configurations are often bloated; the goal is to simplify the logic using Cloudflare's modern primitives. For large enterprise migrations (100+ zones), Terraform or API-driven scripts are used instead of the dashboard to ensure consistency and rollback.

---

## What Students Leave With

- A working conceptual model of Cloudflare's application-security order of operations

- Hands-on experience across ~55 lab exercises spanning DNS, SSL, CDN, traffic management, load balancing, WAF, threat intelligence, Bot Management, Rate Limiting, API Shield, and Page Shield
- A completed individual capstone that takes an exposed origin to a sealed, validated perimeter, checked by an automated grader
- The judgment to select the right tool for the right traffic profile, since no single product protects against everything

---

## Cloudflare Products Covered

| Product | Course Coverage |
|---|---|
| **DNS** | Full and partial (CNAME) setups, CNAME flattening, proxy status, port limitations |
| **SSL/TLS** | Universal, Advanced, and custom certificates; the five SSL modes; Full (Strict) as best practice; DCV records; auto-renewal |
| **CDN** | Default caching, cache-control directives, `cf-cache-status`, Tiered Caching, cache keys, TTL strategy, cache-purge safety |
| **Argo Smart Routing** | Dynamic-traffic acceleration; when it helps and when it doesn't |
| **Rules** | Redirects, header manipulation, Host Header Override, geo-redirection, URL rewrites, Managed Transforms; Last-Match-Wins vs. Top-Down semantics |
| **Load Balancing** | Pools, monitors, steering policies, session affinity, content-aware health checks, health-check pitfalls |
| **DDoS Protection** | Automated volumetric and signature-based mitigation, threshold behavior |
| **WAF Managed Rules** | 728+ rule ruleset, logging-first workflow, zero-day protection |
| **WAF OWASP Ruleset** | Anomaly scoring, paranoia levels, compliance-only enablement guidance |
| **Security Analytics** | Events dashboard, Ray ID filtering, sampling limits, SIEM Logpush |
| **Custom Rules** | Match-and-action model, challenge types, Skip action, top-down execution |
| **Threat Intelligence / Managed Lists** | Preemptive blocking of anonymizers, open proxies, and botnet C2 with zero maintenance |
| **Bot Management** | Scoring (1-99), verified bots, three-step implementation, rule stacking |
| **Rate Limiting** | Match criteria (incl. "IP with NAT support"), threshold selection, tiered stacking |

| | |
|---|---|
| **API Shield** | Discovery + Endpoint Management, Schema Validation, fallthrough rules, mTLS, integrated Rate Limiting, Sequence Analysis |
| **Page Shield** | Client-side resource monitoring, malicious-script detection, Content Security Policy allowlisting |
| **Cloudflare Spectrum** | Mentioned for non-standard-port proxying (context) |

---

## Instructor

Mark Bowling Cloudflare Customer Success

---

*Course materials developed 2026. Product names and dashboard paths reflect the current Cloudflare Application Security dashboard.*