Protect Foundations - Best Practices

PingOne Protect

Field	Value
Version	1.0
Date	2026-04-01
Owner	Partner Delivery Architects
Intended Audience	Technical Consultants/Project Managers
Distribution	Internal/Partner

Related Delivery Kit Assets

- **Protect Foundations - Getting Started**
- **Protect Foundations - Fundamentals**
- **Protect Foundations - PingFederate Integration Guide**
- **Protect Foundations - DaVinci Integration Guide**
- **Protect Foundations - PingAM / AIC Integration Guide**
- **Protect Foundations - Delivery Roadmap Template**
- **Protect Foundations - Delivery Playbook**

Table of Contents

Protect Foundations - Best Practices

This guide captures practical best practices for deploying and tuning PingOne Protect: how to choose predictors, design risk policies, run in learn mode, interpret dashboards, and roll out enforcement safely without under-protecting high-risk flows.

This guide focuses on tuning and optimisation. For delivery flow, follow the **Protect Foundations - Delivery Playbook.**

---

1. Design Principles

These principles should guide every Protect deployment, regardless of integration surface.

1. **Start in observe mode, then enforce.**
   - Run with non-blocking actions first (log/observe, advisory decisions).
   - Use the dashboard and audit views to understand behaviour before you block or hard step-up.
2. **Tune with real traffic, not synthetic.**
   - Let predictors train on real production-like traffic before heavy tuning.
   - Synthetic test accounts rarely represent real behaviour patterns.
3. **Separate policies by journey.**
   - Use different policies for registration, authentication, recovery, and high-risk transactions where possible.
   - This keeps tuning scoped and avoids one noisy flow polluting another.
4. **Reduce false positives before chasing more "High" decisions.**
   - Clean up legitimate High events first (allow lists, composite predictors, segmenting flows).
   - Only then explore making policies more aggressive.
5. **Change the smallest thing that solves the problem.**
   - Order of operations:
     1. Fix data and integration issues.
     2. Use allow lists and custom/composite predictors.
     3. Adjust thresholds or scores.
     4. As a last resort, disable predictors for a given policy.
6. **Never hide High risk by policy overrides.**
   - Do not override High → Low at the policy level; prefer more precise tuning to avoid losing critical signals.

2. Risk Response & Mitigation Patterns

Define clear and consistent actions for each risk level and common risk scenarios. Avoid generic branching without defined outcomes.

Baseline Risk Response Model

- **LOW risk**
  - Allow / continue authentication
  - Minimise friction
- **MEDIUM risk**
  - Step-up authentication (e.g. MFA)
  - Apply additional verification where required
- **HIGH risk**
  - Block request OR enforce strong verification
  - Deny where confidence of fraud is high

Scenario-Specific Mitigation Guidance

Use predictor context and `recommendedAction` to refine decisions beyond simple risk level.

Bot / AiTM Detection

- Enforce CAPTCHA or bot mitigation controls
- Block or rate-limit repeated attempts
- Use recommendedAction (e.g. BOT_MITIGATION) where available

VPN / Proxy / Anonymous Network

- Step-up authentication OR deny based on policy
- Consider allow-listing trusted corporate VPNs

High-Risk Transactions

- Require strong step-up (MFA, IDV)
- Deny or delay transaction if risk exceeds acceptable threshold

Account Recovery

- Apply stricter controls than login
- Require strong identity verification or fallback process

Important

- Do not rely solely on LOW / MEDIUM / HIGH
- Each branch must map to a clear outcome or control
- Use `recommendedAction` and predictor outputs wherever available

3. Predictor Selection Best Practices

These recommendations should be used as a starting point and refined based on real customer data and behaviour.

3.1 General Guidance

- Start with a sane baseline:
  - Use the default risk policy for early learn mode unless you have a strong reason not to.
- Enable predictors that match:
  - **Channel**: workforce vs CIAM.
  - **Journey**: registration, login, recovery, transactions.
  - **Data availability**: which attributes and SDK payloads you can reliably provide.

3.2 CIAM vs Workforce – Typical Emphasis

Use this as a starting point; you will adjust per customer.

CIAM (Customer identities)

- Prioritise:
  - **Bot detection**
  - **Email reputation**
  - **New device**
  - **Anonymous network**
  - **IP reputation & velocity**
  - **Traffic anomaly**
- Registration flows: focus strongly on bot/fake account prevention and disposable email detection.
- Login flows: combine behaviour, location, and device predictors to catch ATO without over-challenging good users.

Workforce

- Prioritise:
  - **User-based risk behavior**
  - **User velocity & geovelocity**
  - **User location anomaly**
  - **IP reputation** and **anonymous network**
  - **New device** for WFH / BYOD / unmanaged devices
- Since workforce devices and locations are more predictable, behavioural and geolocation-based predictors are often very powerful.

3.3 Per-Journey Hints

You'll typically end up with at least the following policy "types":

- **Registration Policy**
  - Emphasise:
    - Email reputation (temporary/disposable domains)
    - Bot detection / suspicious device
    - New device
    - Traffic anomaly
  - Be careful not to rely only on IP-based predictors, which can be noisy for consumer traffic.

- **Authentication Policy**
  - Emphasise:
    - User-based risk behavior
    - Geovelocity and user velocity
    - IP reputation / anonymous network
    - New device
  - Common pattern: Low → frictionless or reduced MFA; Medium → step-up; High → deny or strong step-up.

- **Account Recovery Policy**
  - Emphasise:
    - Geovelocity (impossible travel between recent events)
    - IP reputation, anonymous network
    - User velocity and location anomaly
  - Treat High risk more strictly: e.g., deny, route to manual recovery, or require strong IDV.

- **High-Risk Transactions (payments, profile changes, approvals)**
  - Combine:
    - Device posture (managed vs unmanaged)
    - New device
    - Behavioural predictors
    - IP reputation and traffic anomaly
  - Often enforced even if login policy is more permissive.

4. Training Windows & Data Readiness

4.1 Recommended Training Windows

As a rule of thumb:

- **Workforce journeys**:
  - Let models run for **1–3 weeks** on production traffic before major tuning.
- **CIAM journeys**:
  - Let models run for **2–4 weeks**, given higher diversity in user behaviour and devices.

During this period:

- Use the default risk policy or a lightly customised version.
- Avoid aggressive overrides or score changes unless you're clearly drowning in false positives.
- Do not attempt detailed tuning before sufficient data has been collected within these windows.

4.2 Data Quality & Coverage

Before blaming predictors or models:

- Verify you are sending consistent, non-null values for:
  - `userId` / `userName`
  - Client IP (consider X-Forwarded-For / Client-IP headers)
  - User agent
  - Device/SDK payload where appropriate
- For DaVinci / AIC / PF integrations:
  - Confirm SDK payloads (when used) are actually being passed.
  - Validate that any custom attributes used by custom predictors are present in events.

If predictors show frequent "not evaluated" results, check:

- That required inputs are available and correctly mapped.
- That the integration guide's steps for those predictors (e.g., SDK, headers, or third-party feeds) are complete.

5. Risk Policy Design Patterns

5.1 Use Multiple Policies Per Use Case

Avoid a single “mega policy” for everything. Instead:

- Create separate policies for:
  - Registration
  - Authentication
  - Account recovery
  - High-risk transactions
  - (Optionally) mobile-specific channels
- This allows:
  - Independent tuning (e.g., more aggressive on recovery, more lenient on low-value auth).
  - Clearer troubleshooting when issues arise.

5.2 Start from the Default Policy

- Use the default risk policy as your starting point for most flows.
- Make targeted changes:
  - Disable predictors you know are out of scope (e.g., workforce-only ones in a CIAM-only project, or vice versa).
  - Adjust thresholds only after you have evidence from the dashboard.

5.3 Target Distributions (Guidance, Not SLAs)

Many deployments aim for something approximately like:

- **High**: ~1% of events (genuinely suspicious traffic)
- **Medium**: 0.5–2% (borderline or watch list)
- **Low**: the rest (trusted / normal)

This is not a hard rule, but a sanity check:

- If High is near 0%, you may be under-detecting.
- If High is >5–10% in a stable environment, you may have a tuning or data quality problem.

6. Tuning Strategy & Workflow

6.1 Tuning Order of Operations

When facing noise or undesired behaviour:

1. **Fix integration/data issues first.**
    - Ensure IP, user IDs, SDK payloads, and custom attributes are all correct.

2. **Use allow lists and scoping rules.**
    - Add allow lists for:
        - Known office IP ranges, VPNs, and trusted WAFs.
        - Trusted countries or ASNs where appropriate.
    - Use custom or composite predictors to create more specific conditions instead of blunt overrides.

3. **Use staging policies for risky changes.**
    - When Protect is already influencing decisions:
        - Clone the current policy as a **staging policy**.
        - Route **staging evaluations** in parallel and compare outcomes in the dashboard.
        - Promote the changes once you are confident.

4. **Adjust predictor scores and thresholds.**
    - Lower a predictor's contribution if it's consistently causing noisy highs and allow lists/composites aren't enough.
    - Avoid dropping its contribution to near zero unless you're effectively deciding not to use it.

5. **Disable predictors as a last resort.**
    - If a predictor is systematically misaligned with a specific use case and can't be tuned safely, consider turning it off for that policy.

Always validate changes against real traffic using the dashboard before promoting them to production.

6.2 Handling False Positives

For a cluster of false positives (legitimate users flagged as High):

1. Identify common patterns:
    - Same IP ranges, ASN, country, device type, or traffic source.
    - Same application or subset of journeys.

2. Decide which layer to adjust:
    - **Integration layer** (e.g., correct IP headers, fix SDK deployment)
    - **Policy layer** (scores/thresholds)
    - **Custom predictor/allow list layer** (whitelist certain networks or device types)

3. Document changes:
    - Record what you changed, why, and which metrics or dashboards you used to verify improvement.

6.3 Avoiding Dangerous Overrides

Avoid:

- Global overrides that downgrade High → Low at policy level.
- Changes that make High extremely rare without evidence that risk truly dropped.
- Blindly disabling predictors you don't fully understand.

Instead:

- Prefer more targeted changes:
    - Narrow predictor scope (e.g., limit to certain flows).
    - Use allow lists for known safe cases.
    - Only reduce scores enough to fix the noise, not to hide all High outcomes.

7. Monitoring & KPIs

Monitoring is essential to validate tuning decisions and detect regressions over time.

7.1 Using the Threat Protection Dashboard Effectively

On a regular cadence (daily in early phases, then weekly):

- Check:
    - Overall event counts and risk distribution over time.
    - Top predictors contributing to High / Medium.
    - Top IPs, countries, and devices involved in suspicious behaviour.
- Use drill-down:
    - Click through to individual events and predictors.
    - Capture Resource IDs for deeper analysis in audit logs.

7.2 Suggested KPIs

Depending on the engagement, track:

- **High-risk event rate** (per journey)
    - “Are we landing roughly where we expect?”
- **Medium-risk event rate**
    - “Is Medium meaningful, or just a dumping ground?”
- **False-positive rate**
    - Sample High/Medium events and estimate what percentage are legitimate users.
- **Challenge rate vs completion rate**
    - For flows with MFA or other step-ups:
        - What proportion of events are challenged?
        - Of those, how many succeed vs abandon?

7.3 Operational Dashboards & Alerts

Longer term, surface Protect metrics into:

- Central monitoring (e.g., SOC dashboards, SIEM).
- Alerts for:
    - Sudden spikes in High risk.
    - Sudden drop in overall evaluations (possible integration failure).
    - Predictors consistently failing or returning “not evaluated”.

8. Rollout Patterns & Safety Nets

8.1 Phased Rollout

A safe pattern for most customers:

1. **Phase 0 – Dark / Learn Mode**
    - Integrate Protect into flows with non-blocking actions (e.g., log-only or advisory).
    - Train predictors for the recommended window.
    - Confirm data quality and initial distributions.

2. **Phase 1 – Soft Enforcement**
    - Apply stronger actions for *some* high-risk signals:
        - E.g., enforce MFA or additional steps only for clear High events.
    - Keep Medium mostly advisory or light step-up.
    - Monitor user impact and adjust.

3. **Phase 2 – Full Enforcement**
    - Promote proven policies to production.
    - Enforce High = deny/deflect in high-value flows.
    - Continue to refine predictors and allow lists as new patterns appear.

8.2 Rollback & Failsafe

Always define:

- **Rollback plan**:
    - Ability to switch back to a previous policy version or staging policy.
    - Quick ways to relax decisions (e.g., temporarily treating High as Medium) if necessary.

- **Change windows**:
    - Apply major tuning changes within a controlled window with monitoring in place.

- **Communication plan**:
    - Inform stakeholders (support, SOC, business owners) ahead of impactful changes.

9. Governance & Ownership

9.1 Roles

Clarify who owns:

- **Policy design & tuning**
  - Typically the IAM / security architecture team with input from fraud / risk.

- **Operations & monitoring**
  - Platform / IAM operations, potentially SOC for high-risk alerts.

- **Change review**
  - A small decision group (e.g., IAM architect + security owner) reviewing and approving policy changes that affect user experience or risk posture.

9.2 Change Management

For each change:

- Record:
  - What was changed (policy, predictor, scores, allow lists).
  - Why (false positives, missed risk, new use case).
  - Expected impact and rollout plan.
- After deployment:
  - Review dashboard and key KPIs after an agreed period.
  - Decide whether to keep, refine, or roll back.

9.3 Alignment with Broader Controls

Ensure Protect decisions align with:

- **Access policies** in PingOne / applications.
- MFA / passwordless strategies (e.g., when to skip MFA vs when to insist).