AppTesting Product Overview

AppTesting Product Overview

> *Read-Only Application Analysis, Testing & Security Insights for Microsoft Entra™ ID*

AppTesting is the read-only analysis and troubleshooting component of the **AppConfig² Suite**. It delivers the full depth of application visibility, authentication flow testing, token analysis, security posture assessment, and Microsoft Graph data exploration—while enforcing a strict **no-change guarantee**. This makes it ideal for regulated environments, production operations, security review processes, and organizations with enforced change control policies.

🔍 Positioning & Purpose

Where AppConfig provides full configuration, provisioning, lifecycle and recovery capabilities, **AppTesting focuses exclusively on safe exploration, diagnostics, and insight generation**. It's built for:

- Security, support, and engineering teams that need to investigate issues without risk of accidental modification

- Enterprises with strict separation of duties (SoD) or ITIL-based change management
- Environments where production tenant changes must occur only via Microsoft Entra Portal or controlled pipelines
- Pre-implementation and audit review workflows

Goal	Use AppTesting When	Use AppConfig When
Investigating authentication failures	✅	✅
Decoding & analyzing tokens	✅	✅
Viewing applied Conditional Access policies	✅	✅
Assessing permission exposure & attack surface	✅	✅
Running OAuth test flows & Graph queries	✅	✅
Modifying application configuration	❌	✅
Creating / editing app roles or claims policies	❌	✅
Generating client secrets / credentials	❌	✅
User provisioning & directory extensions	❌	✅
Backup, restore, lifecycle management	❌	✅

🧭 Core Principles

1. Safety First – Cannot alter applications, credentials, permissions, redirect URIs, claims, roles, or exposed API settings.
2. Deep Insight – Surfaces operational, security, and structural data otherwise scattered across multiple Entra and Graph experiences.
3. Accelerated Troubleshooting – Combines authentication flow testers, token tools, permission analysis, and Graph explorer in one workspace.
4. Compliance Friendly – Enables regulated teams to participate in security and readiness work without policy exceptions.
5. Future-Proof – Aligned with evolving Microsoft Entra ID patterns and Graph API surface.

🌟 Key Capabilities (Read-Only)

Authentication & Protocol Diagnostics

- Multi-flow OAuth2/OIDC testing (Auth Code, PKCE, Client Credentials, hybrid)
- Raw OAuth URL generation & response inspection (without MSAL abstraction)
- Token Scope Requester for delegated & application tokens
- State, nonce, and parameter validation helpers

Token Analysis & Claims Intelligence

- JWT decoding with structured visual claim breakdown
- Claims mapping visibility (where policies affect output tokens)
- Optional claims presence confirmation
- Expiration, signature, audience, and scope validation cues

Security & Exposure Analysis

- Permission (API scope / app role) inventory with risk scoring heuristics
- Attack surface summarization (high privilege exposure vectors)
- Certificate & secret expiration monitor (read-only visibility)
- Conditional Access insight surface (impact-focused context)

Application Portfolio Visibility

- Application classification & ownership details
- Service principal relationships & assignments overview
- Lifecycle metadata (creation, ownership, usage indicators)
- Export-ready tabular data for audit & review workflows

Embedded Graph Explorer (Safe Mode)

- Pre-built request templates for common diagnostic flows
- Response formatting & JSON inspection
- Read-only guardrails: only permitted GET operations (where enforced)
- OData Query Builder to shape and filter dataset exploration

Suite Diagnostic Tools

- Token Scope Requester
- Raw OAuth Tester
- OData Query Builder
- Secrets & Certificate Expiration Monitor (visibility only)

🔐 What AppTesting Explicitly Does NOT Do

Not Included	Rationale
Create, edit, or delete applications	Ensures zero-risk operational posture
Modify redirect URIs or identifiers	Prevents outage-inducing misconfiguration
Generate or revoke secrets & certificates	Avoids credential proliferation risk
Create or edit App Roles	Preserves least privilege governance
Assign or modify API permissions	Keeps permission acquisition in approved workflows
Configure exposed APIs / scopes	Ensures published APIs remain controlled
Apply or edit claims mapping policies	Restricts token surface alteration
Manage directory extensions	Limits schema changes to controlled admins
Perform backup / restore operations	Reserved for full management tool (AppConfig)

🆚 AppTesting vs AppConfig – Architectural Contrast

Dimension	AppTesting	AppConfig
Change Scope	None (read-only)	Full (create, modify, restore)
Risk Profile	Zero configuration impact	Managed & reversible changes
Ideal Audience	Security, Support, Audit, Governance	Engineering, Identity Ops, Platform owners
Backup & Restore	View only outcomes / state	Automatic silent backup & one-click restore
Claims & Roles	Visibility only	Full lifecycle management
Credential Management	Inspect only	Generate & rotate
API Exposure	Observability	Configure & publish
Directory Extensions	Read-only	Create & manage
Lifecycle Operations	Observe	Create / retire / recover

🎯 Common Scenarios

Scenario	Why AppTesting Fits
Incident response: suspicious app behavior	Rapid, safe analysis without risk of altering evidence
Pre-change review of production apps	Confirms current state before approved changes via AppConfig or Portal
Security posture audit	Centralizes permission, exposure, and attack surface insights
Support ticket triage	Quick investigation of misconfig claims or token anomalies
Read-only vendor / consulting access	Grants deep insight without configuration authority
Compliance oversight boards	Enables structured review without policy exceptions

🏁 Getting Started

1. Deploy the AppConfig² Suite from Azure Marketplace (trial available).
2. Sign in and select **AppTesting** from the launch interface.
3. Locate the target application via search or portfolio filters.
4. Run authentication and token diagnostics.
5. Review permissions, attack surface, and supporting Graph data.
6. Export findings for ticketing, audit, or remediation planning.
7. (Optional) Escalate to AppConfig for approved configuration changes.

🔄 Upgrade Path to AppConfig

When remediation or structural changes are required:

- Launch AppConfig (full management counterpart)
- Perform controlled configuration updates (roles, claims, redirect URIs, credentials, API exposure)
- Rely on automatic backups & restore if rollback needed
- Feed improved state back into AppTesting for verification

🧪 Governance & Compliance Alignment

- Supports separation of duties (SoD) by isolating diagnostic vs modification functions
- Reduces need for privileged elevation during investigations
- Minimizes blast radius in production-tier tenants
- Provides immutable observation layer supporting audit trails

❓ FAQ (Focused)

Q: Can AppTesting ever accidentally change an app?
A: No. Configuration changes are physically excluded from the tool surface.

Q: Do I need both tools?
A: Many organizations deploy both—AppTesting for daily safe analysis, AppConfig for controlled change windows.

Q: Is there a performance impact on tenants?
A: Data retrieval is optimized and read-only; no write contention or replication delays are introduced.

Q: How does AppTesting help security teams?
A: It centralizes exposure vectors (permissions, roles, certificate expirations) and reduces time-to-insight.

📨 Support

Need help or have feedback?
Email: **support@appconfig.app**
LinkedIn: https://www.linkedin.com/company/appconfig-square
GitHub: https://github.com/AppConfig-Org/AppConfig-Squared

AppTesting – Insight without risk. Part of the AppConfig² Suite.