AMTSO Standard Compliance Statement

This Test has been designed to comply with the Anti-Malware Testing Standards Organization, Inc. ("AMTSO") Testing Protocol Standard for the Testing of Anti-Malware Solutions, Version [1.3] (the "Standard"). This Test Plan has been prepared using the AMTSO Test Plan Template and Usage Directions, Version [2.4]. SecureIQLab is solely responsible for the content of this Test Plan.

SecureIQLab

Methodology

Cloud Native Firewall CyberRisk Validation Methodology

Version: 1.0

Last Revision: 26th March 2026

Language: English

TABLE OF CONTENTS

1. INTRODUCTION

As enterprises continue their cloud journey, maintaining robust security controls becomes both more critical and more complex. In response, many organizations, especially those with stringent regulatory requirements or mission-critical workloads are adopting Cloud-Native Firewalls (CNFWs) to extend trusted security architectures into dynamic cloud environments.

1.1 DEFINITION

A Cloud-Native Firewall (CNFW) is a purpose-built, API-integrated, and centrally managed firewall that acts as a dedicated security enforcement plane for cloud workloads. CNFWs apply various inspection techniques including stateful packet inspection, Layer 7 application-aware filtering, and behavioral anomaly detection to reduce organizational risk across cloud environments. They shift the trust boundary from the physical network perimeter to the cloud control plane, leveraging embedded security engines, dynamic policy management, and native integration with cloud identity and orchestration services. By operationalizing Zero Trust principles directly at the workload layer, CNFWs secure north-south traffic, east-west lateral traffic between microservices, and access to private applications across managed VMs, containers, serverless functions, and hybrid environments.

Why Evaluate Cloud-Native Firewalls Thoroughly?

While many vendors market Cloud-Native Firewall (CNFW) solutions with promises of broad threat coverage and simplified, automated management, real-world performance and security efficacy can vary significantly. Native CNFW offerings though tightly integrated into cloud platforms may lack deep inspection for east-west container traffic, offer limited Layer 7 controls, or depend on static rule sets without advanced threat intelligence integration. Conversely, some third-party CNFWs particularly those using sidecars, daemonsets, or service mesh hooks may introduce latency, resource overhead, or compatibility challenges within Kubernetes and cloud-native environments.

1.2 CLOUD-NATIVE VS. TRADITIONAL FIREWALL DISTINCTIONS

While traditional on-premises firewalls and cloud-native firewalls share foundational security objectives, their architectures and operational models diverge significantly.

A. **Traditional firewalls** rely on hardware appliances or VM-based proxies deployed at fixed network perimeters, with static rule sets and manual provisioning workflows.
B. **Cloud-Native Firewalls** embed security enforcement directly into the cloud control plane, supporting API-driven policy management, auto-scaling enforcement, and dynamic identity-aware access based on cloud IAM roles, resource tags, and Kubernetes ServiceAccounts.

CNFWs enable contextual access enforcement, east-west container traffic inspection, integration with DevSecOps pipelines, and seamless multi-cloud policy management. This distinction redefines the firewall not just as a perimeter control but as a strategic enforcement layer within the enterprise Zero Trust cloud architecture.

1.3 CLOUD THREAT LANDSCAPE & EMERGING RISKS

The cloud's centrality to enterprise operations makes it a prime target for adversaries. Key risks include:

- **API Abuse & Cloud Metadata Exploitation:** Attackers exploit misconfigured APIs and cloud instance metadata services to harvest credentials and escalate privileges.
- **Container Breakouts & Runtime Exploitation:** Vulnerabilities in container runtimes (e.g., runc) enable attackers to escape sandboxes and compromise host infrastructure.
- **Lateral Movement Across Cloud Services:** Compromised microservices or IAM roles enable pivoting between VPCs, accounts, and cloud-native services.
- **Data Exfiltration via Cloud Storage Misconfigurations:** Overly permissive S3, Azure Blob, or GCP Cloud Storage policies expose sensitive data to unauthorized access.
- **Supply Chain Threats:** Malicious container images or compromised build pipeline dependencies introduce backdoors into production workloads.
- **Emerging Risks (GenAI Workload Abuse):** Sensitive enterprise data submitted to GenAI inference endpoints or exfiltrated through prompt injection targeting cloud-hosted LLM APIs.

1.4 ENTERPRISE ADOPTION DRIVERS

Organizations across industries are adopting Cloud-Native Firewalls to address:

- **Cloud-first and hybrid work realities:** Enterprises require consistent security enforcement across multi-cloud and on-premises environments.
- **DevSecOps integration demands:** Security teams need firewalls that integrate with IaC tooling (Terraform, CloudFormation, Helm) and CI/CD pipelines.
- **Container and Kubernetes adoption:** Dynamic, ephemeral workloads require policy enforcement that adapts automatically as workloads scale, move, or restart.
- **East-west traffic visibility:** Organizations need inspection of lateral traffic between microservices a blind spot for traditional perimeter firewalls.

1.5 RATIONALE FOR INDEPENDENT CYBER RISK VALIDATION

As the Cloud-Native Firewall market matures, vendors are introducing varied architectures and claims ranging from cloud-provider-native managed services to third-party containerized agents and FWaaS offerings. Enterprises require clear, objective evidence of how these solutions perform under real-world conditions. Independent cyber risk validation provides:

- **Vendor-neutral benchmarking** of security efficacy, operational effectiveness, and compliance readiness across cloud platforms.
- **Reproducible, transparent test methodologies** aligned with enterprise risk management frameworks including MITRE ATT&CK (Cloud Matrix), STRIDE, OWASP Cloud-Native Guidelines, and CSA CCM.
- **Actionable insights** that support procurement decisions, cloud architecture strategies, and return on security investment assessments.

2. SCOPE AND OBJECTIVES

2.1 SCOPE

This validation methodology applies to Cloud-Native Firewalls (CNFWs) which are purpose-built security solutions that enforce network traffic policy for cloud workloads without relying on physical appliances or manually provisioned VM-based proxies. The scope of this validation includes:

- **Deployment Models:** Managed cloud-provider-native firewall services (e.g., AWS Network Firewall, Azure Firewall, GCP Cloud Firewall) and third-party containerized or agent-based firewalls for Kubernetes environments.
- **Cloud Environments:** Public cloud (AWS, Azure, GCP), hybrid-cloud, multi-cloud, Kubernetes-based (EKS, AKS, GKE), and serverless workload environments.
- **Traffic Types:** North-south (internet-to-cloud) and east-west (intra-cloud, intra-cluster) traffic inspection and enforcement.
- **Core Validation Domains:** Security efficacy, Operational Efficiency, and compliance efficacy.

This methodology does not attempt to validate unrelated endpoint security or traditional on-premises network security solutions:

- Physical hardware firewall appliances (on-premises only)
- Web Application Firewalls (WAF) covered in a separate SecureIQLab WAF methodology
- Endpoint-based host firewalls (Windows Defender Firewall, iptables on individual hosts)
- CASB (Cloud Access Security Broker) solutions without network firewall functionality

2.2 OBJECTIVES

The objectives of this Cloud-Native Firewall validation are to provide enterprises, regulators, and vendors with an independent, repeatable, and transparent framework for evaluating CNFW solutions. Specifically, this methodology seeks to:

A. Quantify Security Efficacy

B. Validate Operational Efficiency & Cloud Readiness

C. Assess Compliance Efficacy

D. Enable Vendor-Neutral Comparisons

3. CYBERRISK VALIDATION CRITERIA

Cyber Risks Validation Criteria for the Cloud-Native Firewall (CNFW) validation methodology are organized into three pillars: Security Efficacy, Operational Efficiency, and Compliance Validation. Each pillar is defined by testable validation criteria that measure how effectively the Cloud-Native Firewall:

- Mitigates cloud-borne threats, lateral movement, and data exposure risks (Security Efficacy),
- Supports operational efficiency and integration within enterprise cloud and DevSecOps environments (Operational Efficacy), and
- Enables alignment with regulatory, audit, and governance obligations (Compliance Efficacy).

3.1 SECURITY EFFICACY

This structure ensures that validation extends beyond theoretical capabilities to practical, measurable outcomes.

3.1.1 THREAT DEFENSE

A. Application-based Threats:

The Cloud-Native Firewall must detect and block threats exploiting web applications, APIs, and cloud-native services, including:

- **Injection Attacks**: SQL Injection, Command Injection, and OS command injection targeting cloud application tiers.
- **Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF)**: Attacks targeting browser-facing cloud applications.
- **Server-Side Request Forgery (SSRF)**: Exploiting cloud metadata API endpoints (AWS IMDSv1/v2, Azure IMDS, GCP metadata server) to harvest credentials.
- **API Abuse & Malicious Payload Injection**: Malformed or malicious JSON/XML payloads targeting microservice APIs.
- **Application-Layer Denial of Service (DoS)**: Volumetric and logic-based attacks disrupting cloud application availability.
- **Known CVE Exploit Blocking**: Using IPS signatures and anomaly detection to block exploitation of publicly disclosed vulnerabilities.

B. Vulnerability-Based Attacks:

The Cloud-Native Firewall must detect and block attempts to identify and exploit unpatched vulnerabilities or misconfigurations, including:

- **Known CVE Exploit Blocking**: Dropping exploit payloads targeting high-risk CVEs and preventing post-exploit actions such as reverse shells and code execution.
- **Automated Vulnerability Scanning Detection:** Identifying and mitigating both credentialed and non-credentialed vulnerability scans, blocking or throttling suspicious probe traffic.
- **Zero-Day Heuristic & Anomaly Detection**: Flagging or blocking suspicious traffic patterns that do not match any known signature, preventing signature-gap exploits.

C. Malware & Botnet Defense:

The Cloud-Native Firewall must detect, block, and contain malware threats and botnet command-and-control activity, including:

- **Malware Download over HTTPS**: Detecting and blocking polymorphic or obfuscated malware served via encrypted channels, including assessment of TLS inspection blind spots.
- **Compressed Malicious Files**: Unpacking and inspecting nested archives (e.g., .zip within .rar) and encrypted archives to detect embedded malicious payloads.
- **Botnet C2 Traffic Simulation**: Identifying and blocking command-and-control communications with varying beacon intervals, DNS-based C2 channels, and fast-flux DNS infrastructure.
- **DNS over HTTPS (DoH) & Encrypted DNS Tunneling**: Detecting malicious C2 traffic concealed within encrypted DNS requests.
- **Time-Delayed or Trigger-Based Payloads**: Validating the firewall's ability to detect threats that activate after a specified time or trigger condition.

D. Browser-Based & Web-Delivered Threats:

The Cloud-Native Firewall must detect and block threats originating from web interactions targeting cloud-hosted assets, including:

- **Browser Exploits**: Deep HTTP/HTTPS inspection to detect and block malicious payloads exploiting browser-engine vulnerabilities.
- **HTML Smuggling**: Detecting client-side payload construction via JavaScript blob assembly and anomalous script behavior that writes or executes files in the browser context.
- **Obfuscated JavaScript**: Blocking heavily obfuscated scripts using multiple encoding steps, runtime eval() constructs, or high-entropy string concatenations.
- **Plugin & Renderer Exploits**: Detecting and blocking exploits targeting PDF renderers and embedded media players using IPS signatures or heuristic analysis.

E. Data Loss & Cloud Storage Leakage:

The Cloud-Native Firewall must detect and prevent unauthorized data exfiltration and enforce data protection policies across cloud environments, including:

- **Bulk Data Exfiltration**: Detecting large-scale transfers of sensitive information to external FTP, HTTP upload endpoints, or unauthorized cloud storage buckets.
- **DLP Violations (PII, PHI, PCI)**: Identifying and blocking transmission of personally identifiable information, protected health information, or payment card data from cloud workloads to unauthorized destinations.
- **Misconfigured Cloud Storage Leaks**: Detecting and alerting on overly permissive access controls on AWS S3, Azure Blob Storage, or GCP Cloud Storage enabling unauthorized data access.
- **Cross-Cloud Exfiltration**: Monitoring and blocking sensitive data transferred across cloud platforms (e.g., AWS to Azure) using built-in cloud connectors.
- **Insider Threat & Privileged Abuse**: Detecting authorized users copying or moving confidential data to unauthorized external endpoints or personal cloud accounts.

F. Container & Serverless Security:

The Cloud-Native Firewall must provide deep container-aware inspection, integrate with orchestration layers (Kubernetes, Docker), and enforce behavioral and policy-based protections, including:

- **Serverless Function Exploit Blocking**: Detecting and halting malicious HTTP(S) requests targeting Lambda, Azure Function, or GCP Function endpoints, including SSRF payloads directed at cloud metadata services.
- **Supply-Chain Threat Mitigation**: Preventing containers or functions from pulling images or dependencies from untrusted registries, and quarantining artifacts flagged by integrated vulnerability-scanning services.
- **Container Runtime Exploitation Detection**: Identifying attempts to exploit container runtimes (e.g., runc, Dirty COW) or host-kernel vulnerabilities, blocking post-exploit network activity such as reverse shells or SSH tunnels.
- **Privileged Container Misuse**: Detecting pods using excessive or dangerous capabilities (privileged, hostPath, Docker socket mounts) and blocking subsequent unauthorized network egress or service discovery.
- **Malicious Container Image Detection**: Identifying and blocking deployment of container images flagged by image-scanning integrations or threat intelligence feeds.

G. GenAI Workload Security:

The rapid adoption of GenAI workloads, inference endpoints, and AI agent frameworks within cloud environments introduces a new class of network-layer security risks that Cloud-Native Firewalls must be capable of detecting and enforcing policy against.

- **Inference Endpoint & Data Protection**
  - Sensitive Data Submission to GenAI Inference APIs
  - Prompt Injection via Cloud LLM APIs
  - Unauthorized GenAI Service Access
  - Unauthorized AI Resource Consumption
- **MCP Server Security**
  - Unauthorized MCP Server Access Control
  - Data Exfiltration via MCP Tool-Call Responses
  - Lateral Movement Through MCP Tool Chains
  - MCP Tool Hijacking via Prompt Injection
  - Credential & Secret Leakage via MCP Responses

3.1.2 Policy Enforcement

The Cloud-Native Firewall must provide granular, identity-aware, and context-driven policy enforcement capabilities. Key requirements include:

A. Stateful Inspection: A CNFW with stateful inspection must demonstrate enforcement via the 5-tuple method (Source IP, Destination IP, Source Port, Destination Port, State/Context), maintaining session state across cloud workload scaling events.

B. Application Control: The CNFW must identify and enforce allow/deny policies for cloud-relevant applications including productivity suites (Google Workspace, Microsoft 365), collaboration platforms (Slack, Teams, Cisco Spark), CRM and ERP tools (Salesforce, ServiceNow), cloud storage services (Google Drive, OneDrive), and DevOps platforms (Jira, GitHub, DocuSign).

C. Geolocation Control: The CNFW must enforce geolocation-based allow/deny policies covering high-risk countries (e.g., Belarus, Cuba, Iran, Iraq, North Korea, Syria, Russia, China, Zimbabwe, Myanmar) and selectively permitting trusted regions as required by organizational policy.

D. Service & Port Control: The CNFW must enforce allow/deny policies for specific services and port/protocol combinations including HTTP (80), HTTPS (443), MySQL (3306), PostgreSQL (5432), SSH (22), SMTP (25), DNS (53), Redis (6379), Elasticsearch (9200), Kafka, RabbitMQ (5672), Jenkins (8080), Prometheus (9090), and OpenLDAP (389).

E. Web/URL Filtering: The CNFW must provide granular URL category filtering across content and service categories (information resources, media, e-commerce, financial, developer platforms, healthcare), security and technology risk categories (malware, command-and-control, newly registered domains, hacking, grayware, ransomware, encrypted DNS), and AI-specific categories (conversational AI, generative search engines, assistive/co-pilot AI, creative studio AI).

F. IP/Port Control: The CNFW must enforce policy for specific IP address and port pairs, including dynamic enforcement based on cloud workload identity and resource tags rather than static IP assignment alone.

G. TOR Exit Node Control: The CNFW must identify and enforce allow/deny policy for connection attempts originating from or destined for known TOR exit nodes, leveraging dynamic threat intelligence feeds.

H. Identity-Aware Policies: The CNFW must enforce context-aware security policies based on cloud identity context including IAM roles, Kubernetes ServiceAccounts, resource tags, organizational hierarchy (organization, project, VPC level), and workload identity rather than relying solely on static IP-based rules.

3.1.3 Advanced Evasive Techniques

The Cloud-Native Firewall must detect and block sophisticated evasion methods used by adversaries to bypass security inspection. Validation covers:

A. Protocol Tunneling & Port Hopping: Embedding malicious traffic within legitimate protocols (HTTP, DNS) or dynamically switching ports to evade signature-based detection.

B. Encrypted or Obfuscated Payloads: Malicious files or commands compressed, encrypted, or encoded to avoid detection, including password-protected archives and steganography.

C. Fragmentation & Split Attacks: Splitting malicious payloads across multiple packets, segments, or sessions to prevent payload reconstruction by inspection engines.

D. Living off the Land (LotL): Leveraging built-in cloud tools or OS utilities that appear legitimate to evade signature-based detection, detectable only through behavioral analytics.

E. Multiple Layers of Encryption / Double-Encapsulation: Simultaneous use of multiple encryption protocols (TLS over SSH) or multi-hop routes that complicate traffic inspection.

F. Evasive Command-and-Control (C2): The use of domain-generation algorithms, frequent IP rotation, or legitimate cloud services (GitHub, cloud storage) as C2 proxies.

G. Polymorphic & Metamorphic Payloads: Malware that alters its binary signature (polymorphic) or entirely rewrites its codebase (metamorphic) between iterations to defeat static and behavioral analysis.

H. Timing-Based Evasion: Intentionally delaying or slowing packet delivery to evade rate-threshold detection systems; includes Slowloris-style attacks, slow beaconing, and extended TCP fragment timeouts.

I. Protocol Evasion: Exploiting protocol parser ambiguities via overlapping TCP fragments, double-encoded HTTP payloads, malformed DNS queries, and non-canonical URL paths.

J. Traffic Padding & Randomization: Injecting randomized junk data alongside real traffic to disrupt flow-based anomaly detection and conceal C2 channel patterns.

K. Advanced Persistent Threats (APTs): Cloud-targeted state-sponsored and cybercrime campaigns (e.g., Cloud Hopper, Silent Fade, Operation Cloudy Omega) leveraging spear-phishing, zero-day exploits, and lateral movement techniques mapped to MITRE ATT&CK Cloud Matrix TTPs.

3.1.4 Cloud-Centric Post-Exploitation Techniques

The CNFW must detect and block adversary actions following initial access, including:

- Credential & token harvesting: accessing API keys, tokens, and certificates from environment variables, CI/CD pipelines, or cloud metadata endpoints.
- Cloud metadata service exploitation (SSRF to AWS IMDSv1/v2, Azure IMDS, GCP metadata) to obtain temporary credentials and escalate privileges.
- Lateral movement through microservice APIs and internal cloud services to pivot across VPCs, accounts, and data stores.
- Data exfiltration via cloud storage snapshots, replicas, or unauthorized cross-account sync operations.

- Use of alternate authentication material (cloud tokens, OAuth tokens MITRE T1550.003) to pivot without repeated authentication.
- Container-specific post-exploitation: malicious image injection, Docker Remote API exploitation, resource hijacking (cryptojacking), and unauthorized cloud metadata API access.

3.1.5 Encryption Capabilities

Validating encryption capabilities in Cloud-Native Firewalls is a critical necessity for data security, regulatory compliance, and effective threat defense. SecureIQLab evaluates the following encryption-related capabilities:

A. SSL/TLS Cipher Support: To detect threats embedded in encrypted traffic, the CNFW must support TLS inspection across TLS 1.2 and TLS 1.3 cipher suites. The firewall must decrypt packets, inspect content, and enforce policy without degrading performance at enterprise scale. TLS 1.3 eliminates legacy cryptographic algorithms and reduces attack surfaces including downgrade attacks and Man-in-the-Middle (MitM) interception. SecureIQLab will test all 22 TLS 1.2 cipher suites and the three TLS 1.3 cipher suites, including mixed-cipher scenarios between client and server, weak cipher handling, and secure cipher fallback enforcement.
B. TLS Session Reuse: The CNFW must support TLS session reuse mechanisms to minimize latency and computational overhead while maintaining inspection fidelity. Validation covers both Session ID (server-side session caching for reconnecting clients) and Session Ticket (client-side session caching to offload server memory) methods, verifying that these mechanisms operate correctly under concurrent enterprise-scale connection loads.

3.2 OPERATIONAL EFFICIENCY

Operational Efficacy evaluates the Cloud-Native Firewall's enterprise readiness, manageability, and impact on cloud workload performance. It measures whether the solution can scale seamlessly across dynamic cloud environments, enforce security policies without disrupting DevSecOps workflows, and integrate effectively with existing enterprise IT, cloud-native, and security infrastructure.

3.2.1 Deployment & Onboarding

The Cloud-Native Firewall must support efficient, scalable deployment and onboarding across enterprise cloud environments:

- **IaC-Driven Deployment**: Support policy definition and deployment through Terraform, CloudFormation, ARM templates, and Helm charts, enabling version-controlled, repeatable cloud firewall provisioning.
- **Multi-Cloud & Multi-Region Support**: Compatible with AWS, Azure, GCP, and Kubernetes-based environments (EKS, AKS, GKE) across multiple regions and accounts.
- **Automated Policy Propagation**: Automatically enforce security and access policies across cloud workloads without manual intervention during scaling events or deployments.
- **GitOps Integration**: Native integration with GitOps tooling (e.g., Argo CD, Flux) for automated policy rollout and drift detection.
- **Third-Party & Contractor Access**: Onboard external workloads securely without extending full network trust, enforcing identity-based access policies per workload or project context.

3.2.2 Policy Management & Administration

- **Time to Create & Apply Policies:** Rapid policy creation and deployment across cloud accounts, projects, VPCs, Kubernetes namespaces, and workload identities.
- **Policy Propagation Time:** Fast and reliable propagation of policies to managed cloud workloads, container clusters, and serverless function environments.
- **Policy Engine Granularity & Flexibility:** Define rules per IAM role, Kubernetes ServiceAccount, cloud resource tag, application, URL category, or data type.
- **Admin Console Usability:** Intuitive interface with clear dashboards, policy creation wizards, and efficient workflows for monitoring, alerting, and reporting.
- **Granular Role-Based Access Control (RBAC):** Assign differentiated administrative roles with fine-grained privileges across cloud accounts and security domains.
- **Policy Conflict Detection & Resolution:** Identify and manage overlapping or conflicting rules including conflicts between cloud-provider-native policies and third-party CNFW rules before enforcement.
- **Policy Versioning & Rollback:** Maintain version history of IaC-defined policies and revert to previous configurations on-demand.

3.2.3 Integration with Enterprise Ecosystem

The Cloud-Native Firewall must integrate seamlessly with the enterprise security, cloud-native, and IT ecosystem:

- **Cloud Identity Provider (IdP) Integration**: Support IAM-role-based enforcement, SSO, MFA, and RBAC via AWS IAM, Azure Active Directory, GCP IAM, Okta, Ping Identity, and other enterprise IdPs.
- **SIEM/SOAR Integration**: Forward logs, alerts, and events to cloud-native logging pipelines (AWS CloudWatch, Azure Monitor, GCP Cloud Logging) and downstream SIEM/SOAR platforms (Splunk, Microsoft Sentinel, Palo Alto XSOAR).
- **Cloud-Native Logging Pipeline Support**: Native integration with cloud provider logging services for immutable audit trails and automated compliance evidence collection.
- **DLP/EDR/XDR Interoperability**: Share telemetry and enforce security policies across cloud workloads, endpoint, and network layers.
- **API/SDK Support**: Enable integration with custom DevSecOps tooling, CI/CD pipelines, and enterprise IT management platforms.
- **Threat Intelligence Feeds**: Dynamically update policies based on external threat intelligence for malware, C2 domains, malicious IPs, and newly registered domains.

3.2.4 Scalability & Maintenance

- **Auto-Scaling Support**: Demonstrate elastic scaling of firewall enforcement capacity in response to cloud workload demand, with no degradation in inspection fidelity during scaling events.
- **Update Mechanism Type**: Support managed, automated, and staged updates for firewall engines, threat intelligence signatures, and policy configurations.

- **Deployment Scalability**: Rapidly onboard and enforce policy across large numbers of cloud accounts, Kubernetes clusters, and serverless function environments without manual provisioning.
- **Policy Scalability**: Manage thousands of workload identities, resource tags, IAM roles, and policy rules without performance degradation.
- **Stateful Session Persistence**: Maintain session state during horizontal autoscaling events, failovers, and blue-green deployments.

3.2.5 Incident Response and Visibility

- **Time to Detect & Respond to Blocked Action**: Measure how quickly policy violations (e.g., blocked lateral movement, data exfiltration attempt) are surfaced to security administrators.
- **Logging & Alerting Quality**: Ensure logs capture relevant metadata (workload identity, IAM role, cloud account, source/destination IP, action blocked, threat name, timestamp) with sufficient granularity for compliance and investigation.
- **Forensic Retrieval Capability**: Ability to export or reconstruct event chains (flow logs, API call records, audit trails) for post-incident forensic analysis.
- **Threat Classification & Triage**: Classify threats by severity, correlate related events, and surface actionable investigation workflows for security analysts.
- **Time to Prevent (TTP) Measurement**: Measure the interval from initial detection of an unknown threat to active blocking, reporting samples missed within the TTP window.

3.2.6 REPORTING CAPABILITY

The Cloud-Native Firewall must provide comprehensive, actionable reporting to support security operations, compliance, and executive decision-making:

- **Standard Firewall Security Reporting**: Detection and reporting of known threats including malicious URLs, botnets, phishing, compressed malicious files, exploits, script injection, and adware.
- **Advanced Cloud Firewall Security Reporting**: Reporting on advanced threats including APTs, AETs, malware delivered over TLS, obfuscated JavaScript, container exploit events, and cloud-centric post-exploitation activity.
- **Threat Timeline**: Detailed chronological reconstruction of attack progression from initial access through lateral movement and data exfiltration, including relevant indicators of compromise (IOCs).
- **Centralized Logging & Monitoring**: Integration with centralized logging mechanisms (syslog, Splunk, AWS CloudWatch, Azure Monitor, GCP Cloud Logging) capturing source/destination IP, timestamp, protocol, port, and threat name for every policy-relevant event.
- **Return on Security Investment (ROSI) Metrics**: Quantifiable metrics covering security effectiveness, operational efficiency, false positive rates, and total cost of ownership to support procurement and investment decisions.

3.2.7 Security Administration

- **Role-Based Access Control (RBAC)**: Enforce granular administrative roles across cloud accounts and security domains to prevent unauthorized configuration changes.
- **Audit & Compliance Logging**: Capture tamper-proof logs of all administrative actions, policy changes, and configuration updates for forensic and audit purposes.
- **Alerting & Notification Management**: Configure critical event notifications and integrate with SMTP, SIEM, or incident ticketing platforms (ServiceNow, PagerDuty) for timely alerting.
- **Patch & Threat Intelligence Frequency**: Ensure that the CNFW integrates the latest threat intelligence updates and vulnerability patches on a frequent, automated basis without requiring manual intervention.

3.3 COMPLIANCE VALIDATION

The Cloud-Native Firewall must demonstrate its ability to support enterprise compliance mandates, regulatory requirements, and audit readiness across cloud environments. Validation focuses on how effectively the firewall enforces compliance-driven controls, maintains immutable audit trails, and supports compliance certifications.

3.3.1 Data Privacy & Protection

- **GDPR Alignment**: Verify that CNFW capabilities align with Data Privacy and Protection requirements including identity-aware access, least-privilege policies, data loss prevention, encryption enforcement, session controls, phishing and malware defenses, audit logging, data minimization, geo-fencing, and continuous posture validation.

3.3.2 Industry-Specific Security Standards

- **HIPAA**: CNFW evaluation covers strong authentication and MFA integration, role-based and context-aware access control, audit logging and monitoring, TLS 1.3 encryption with Perfect Forward Secrecy (PFS), data loss prevention (copy-paste, upload/download restrictions), content filtering and malware protection, granular session management, incident detection and alerting, and integration with IdP/CASB/DLP/EDR platforms.
- **PCI DSS**: CNFW evaluation covers identity-aware access (SSO, MFA, step-up authentication), contextual and least-privilege policies, session management, DLP controls (blocking screenshot, print, download), TLS enforcement, browser sandbox and isolation, phishing and malicious site protection, audit logging of user and policy enforcement activity, real-time alerting and SOAR integration, BYOD and third-party access controls, geo-fencing, and continuous monitoring.
- **NIST 800-171**: CNFW evaluation covers identity-aware access, contextual least-privilege policies, session controls, DLP (blocking copy-paste, upload/download, print, screenshots), TLS 1.2+/1.3 encryption, browser sandboxing and OS isolation, phishing protection, audit logging, real-time alerting and ITSM/SOAR workflows, controlled BYOD/contractor access, geo-fencing, and continuous enforcement monitoring.

3.3.3 General Security Standards

- **SOC 2**: CNFW evaluation covers identity-aware access (MFA, SSO, conditional access), contextual access policies (geo, device, role, time, risk-based), privileged access and session controls, DLP controls, encryption enforcement, browser isolation and sandboxing, phishing and malicious site blocking, BYOD and third-party access policies, audit logging, SIEM/SOAR integration, patch and vulnerability management, and geo-fencing.
- **ISO/IEC 27001:2022**: CNFW evaluation covers identity-aware access, contextual and least-privilege access, session management, DLP controls, TLS enforcement and secure storage policies, browser sandboxing, phishing protection, audit logging and SIEM/SOAR integration, BYOD and third-party access control, geo-fencing, continuous monitoring, and secure configuration and patching practices.

3.4 SECURE BY DESIGN AND SECURE BY DEFAULT

3.4.1 Secure by Design - Tactics:

- Document conformance to a secure SDLC framework.
- Document cybersecurity performance goals (CPG) or equivalent conformance.
- Responsible use of open-source software and supply chain risk management.
- Vulnerability management and responsible disclosure policy.
- Publish software bills of materials (SBOMs) and a memory-safety roadmap.
- Publish high-level threat models and detailed secure SDLC self-attestations.
- Align with Zero Trust architecture (ZTA) and cloud-native security principles.
- Establish internal security controls and embrace vulnerability transparency.
- Test SIEM and SOAR integration as part of the secure SDLC process.
- Provide regular reports to the board of directors on security program status.

3.4.2 Secure by Default - Tactics:

- Eliminate default credentials and enforce strong authentication by default.
- Implement least-privilege defaults for all cloud firewall roles and service accounts.
- Actively discourage use of unsafe legacy TLS versions and deprecated cipher suites.
- Provide logging at no additional charge and without feature-gating on audit capabilities.
- Create secure IaC configuration templates and publish patching and update statistics.
- Embrace open standards for API-driven policy management and cloud integration.
- Publish aggregate security-relevant statistics and publicly name a Secure by Design executive sponsor.

4. VALIDATION METHODOLOGY

The SecureIQLab Cloud-Native Firewall Cyber Risk Validation is conducted in nine structured phases, each building upon the previous to ensure comprehensive coverage, reproducibility, and actionable outcomes:

4.1 TEST ENVIRONMENT PREPARATION

- Define and provision cloud accounts, regions, virtual networks (VPCs/VNets), Kubernetes clusters, and container registries.
- Deploy dedicated cloud-hosted attack simulation infrastructure for offensive threat scenarios.
- Isolate test environments with segmented cloud networks (baseline vs. offensive testing accounts/VPCs).
- Create representative cloud workload environments to deploy and test the Cloud-Native Firewall including VM-based, containerized, and serverless workload tiers.
- Obtain and activate Cloud-Native Firewall license(s) and provision required IAM roles and service accounts.
- Deploy and configure the CNFW across relevant cloud environments and Kubernetes clusters.

4.2 BASELINE CONFIGURATION

- Configure baseline settings combining vendor recommendations with enterprise cloud security best practices.
- Deploy monitoring, logging, and telemetry infrastructure (cloud-native logging pipelines, SIEM forwarding) for security, operational, and compliance data collection.
- Validate installation, configuration, and functionality with the vendor team, including IaC policy deployment and Kubernetes integration verification.
- Conduct environment readiness verification using a pre-validation checklist aligned with the vendor's recommended enterprise-ready configuration.

4.3 INITIAL SMOKE TESTING

- Select 20% of representative threat payloads and test cases across Security Efficacy categories.
- Execute smoke tests to validate configuration, policy enforcement, TLS inspection, and baseline rule application.
- Create a smoke test scorecard documenting environment configuration, test coverage, results, and exceptions.
- Review the scorecard internally to identify critical issues, misconfigurations, or IAM permission gaps.
- Share the scorecard with the Cloud-Native Firewall vendor team for clarification, remediation, or guidance.
- Apply remediation or configuration adjustments as needed before full-scale testing.
- Confirm environment readiness to proceed to Main Validation Testing.

4.4 MAIN VALIDATION TESTING

- Execute Security Test Cases across all Threat Defense, Policy Enforcement, Advanced Evasive Techniques, and Encryption capability categories.
- Execute Operational Test Cases covering deployment, policy management, integration, scalability, and performance.
- Execute Compliance Test Cases mapped to GDPR, HIPAA, PCI DSS, NIST 800-171, SOC 2, and ISO/IEC 27001:2022 requirements.
- Each test case is executed across three iterations to verify consistency, reproducibility, and reliability of results.

- Capture metrics and observations including packet captures (PCAP files), cloud flow logs, event telemetry, screenshots, and screen recordings.
- Record observations for each test scenario, including pass, fail, and anomaly results.
- Produce a comprehensive, reproducible scorecard capturing the CNFW's performance across Security, Operational, and Compliance Efficacy domains, including per-test score, sample count, and executive summary.

4.5 SCORECARD COMPILATION

- Map results to validation criteria, with each test case clearly indicating which validation domain it contributes to: Security, Operational, or Compliance.
- The internal SecureIQLab team reviews and validates all test results.
- Any discrepancies, anomalies, or gaps are identified and documented.
- Ensures the accuracy, completeness, and reliability of the scorecard prior to vendor engagement.

4.6 DISPUTE RESOLUTION SESSION

- Provide the vendor with a copy of the scorecard, highlighting detailed and executive-level results.
- Discuss potential configuration issues, environmental factors, IAM permission gaps, or feature limitations that may have impacted results.
- If discrepancies are identified, re-run specific test cases to validate or correct the results.
- Capture vendor clarifications, agreed-upon resolutions, and any remaining unresolved issues.
- Obtain formal acknowledgment from stakeholders and vendors that clarified results are accurate.

4.7 SECUREIQLAB STAKEHOLDER REVIEW

- Share the validated scorecard and detailed observations, highlighting Security, Operational, and Compliance Efficacy findings with SecureIQLab leadership.
- Gather input on interpretation of results, potential gaps, or context-specific considerations for cloud-native environments.
- Update observations, scorecards, and residual risk assessments based on leadership input.

4.8 FINAL REPORT PUBLICATION

- Deliver both an individual comprehensive report and a comparative report, including detailed test results, scorecards, and ROSI analysis.

5. CLOUD-NATIVE FIREWALL TEST ARCHITECTURE AND SETUP OVERVIEW

The SecureIQLab testbed is designed to ensure rigorous, multi-cloud, and real-world validation of Cloud-Native Firewall capabilities. As CNFWs are cloud-delivered security enforcement services, the evaluation focuses on cloud workload integration and the quality of north-south and east-west traffic inspection across representative enterprise cloud architectures.

All participating CNFW solutions will be evaluated across multiple cloud environments and workload types including AWS, Azure, and GCP managed firewall services and Kubernetes-based containerized agents (EKS, AKS, GKE) to validate cross-platform compatibility, policy consistency, and performance resiliency.

To replicate enterprise-grade cloud usage conditions, the test environment incorporates a representative mix of workloads:

- Commonly used SaaS and cloud-native applications (productivity, collaboration, CRM, DevOps platforms) to validate policy enforcement under real application traffic.
- Publicly accessible internet destinations to validate URL filtering, threat intelligence enforcement, and TOR exit node controls.
- Private cloud applications and microservices deployed within secured VPCs and Kubernetes namespaces, accessed through the CNFW to assess zero-trust enforcement and east-west inspection.
- Serverless function endpoints (AWS Lambda, Azure Functions, GCP Cloud Functions) to validate cloud-native workload security controls.

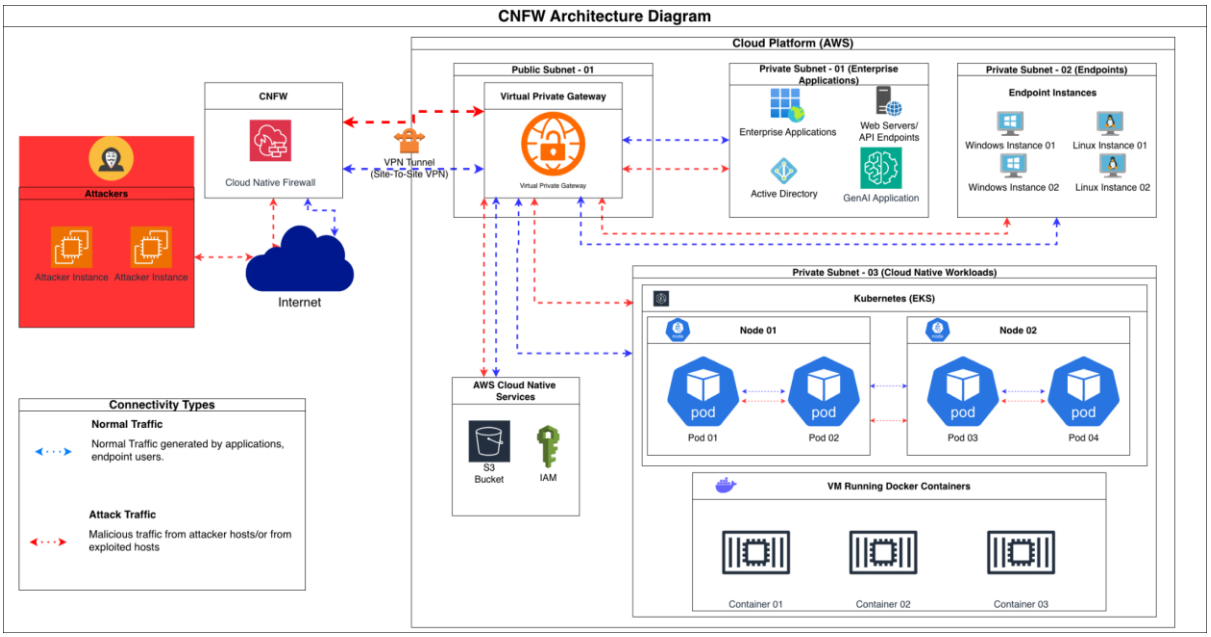

Figure 1. Cloud Native Firewall Validation Architecture

A controlled adversarial simulation environment is provisioned within the cloud lab to deliver targeted attack payloads including container exploits, SSRF attacks against metadata APIs, encrypted C2 traffic, and lateral movement simulations enabling evaluation of the CNFW's detection, blocking, and forensic capabilities.

This architecture ensures a realistic yet controlled validation environment that reflects enterprise cloud-native firewall deployment patterns, while maintaining repeatability, vendor neutrality, and fidelity to modern DevSecOps security operations.

Figure 1 illustrates the Cloud-Native Firewall test architecture, incorporating managed cloud firewall services, containerized enforcement agents, cloud identity integration (IAM/IdP), centralized logging pipelines, and adversarial simulation infrastructure.

6. SCORING MECHANISM FOR CYBER RISK VALIDATION

The scoring mechanism provides a structured approach to assess the effectiveness, operational readiness, and compliance posture of Cloud-Native Firewall (CNFW) solutions. Scores are based on the dual criteria of prevention (ability to stop threats and enforce policy) and detection (ability to log, monitor, and surface security events for audit and response).

Each efficacy dimension Security, Operational, and Compliance has tailored scoring criteria to ensure a holistic evaluation.

6.1 SECURITY EFFICACY SCORING CRITERIA

The Security Efficacy dimension measures how effectively the Cloud-Native Firewall can detect, prevent, and log cyber threats that directly impact organizational cloud security. The goal is to validate whether the CNFW can serve as an effective frontline defense against modern cloud-native attack vectors while ensuring adequate visibility for security operations teams.

The scoring mechanism applies the Prevention + Detection framework to each test case. Effectiveness is graded as follows:

Outcome	Description	Score
Prevent and Detect	The threat is actively prevented and the event is recorded, enabling full auditability and forensic analysis.	**100%**
Prevent, No Detect	The threat is blocked, but without logging there is no visibility for audit or incident response.	**75%**
Detect, No Prevent	The action is allowed but logged, providing post-event visibility without preventative control.	**25%**
No Detect, No Prevent	No prevention and no logging, resulting in a complete security failure.	**0%**

6.2 OPERATIONAL EFFICIENCY SCORING CRITERIA

The scoring criteria offer a structured framework to quantify the operational effectiveness of the Cloud-Native Firewall. They assess key aspects such as cloud-native deployment agility, IaC-driven policy management, multi-cloud scalability, DevSecOps integration, and performance under enterprise workload conditions. Each area is evaluated using a four-tier scale.

Deployment & Onboarding:

Outcome	Description	Score
Seamless & Cloud-Native	Deployment fully automated via IaC (Terraform, CloudFormation, Helm); multi-cloud support across AWS, Azure, GCP, and Kubernetes; GitOps integration; smooth onboarding with IdP/MFA baked in.	**100%**

Functional but Partially Manual	Deployment works across major cloud platforms but requires some manual configuration; GitOps or IaC support incomplete; IdP integration not fully streamlined.	**75%**
Limited & Complex	Deployment covers only a subset of cloud environments; significant manual effort required; IaC or Kubernetes integration absent or unreliable.	**25%**
Unsupported & Fragmented	No cloud-native deployment support; onboarding is ad-hoc, insecure, or fails at scale across cloud environments.	**0%**

Policy Management & Administration:

Outcome	Description	Score
Efficient & Auditable	Policies easy to create via IaC or console; granular enforcement (per IAM role, workload identity, tag, namespace); fast propagation; conflict-aware; version-controlled; auditable.	**100%**
Efficient, No Audit Trail	Policies propagate effectively but lack proper audit/version control, conflict detection, or rollback capabilities.	**75%**
Inefficient but Visible	Policy creation is slow or manual; limited IaC support; propagation lags across cloud accounts and clusters.	**25%**
Inefficient & Opaque	Policies complex to configure; propagation unreliable; no audit, versioning, or rollback visibility.	**0%**

Integration with Enterprise Ecosystem:

Outcome	Description	Score
Fully Integrated	Native support for cloud IdPs (AWS IAM, Azure AD, GCP IAM, Okta); SIEM/SOAR forwarding; cloud-native logging pipelines; DLP/EDR/XDR interoperability; threat intelligence feeds; pre-built connectors.	**100%**
Partially Integrated	Works with major cloud IdPs and SIEMs; limited pre-built integrations; relies on custom API configurations for others.	**75%**

Siloed, Limited Connectors	Minimal IdP/SIEM support; integrations require heavy custom development; cloud-native logging pipelines not natively supported.	**25%**
No Integration	No support for cloud IdPs, SIEM/SOAR, cloud logging APIs, or threat intelligence feeds.	**0%**

Scalability & Maintenance:

Outcome	Description	Score
Highly Scalable & Elastic	Demonstrates elastic auto-scaling of enforcement capacity; supports staged/automated updates; policy and device scalability to thousands of workloads with no degradation.	**100%**
Scalable with Trade-offs	Handles growth but requires staged/manual scaling effort or shows minor performance trade-offs under high-concurrency workloads.	**75%**
Limited Scalability	Performance degrades with larger policy or workload sets; auto-scaling behavior inconsistent; update mechanisms require manual intervention.	**25%**
Not Scalable	Cannot scale enforcement, policy, or updates reliably across enterprise cloud environments.	**0%**

Performance Impact on Cloud Workloads:

Outcome	Description	Score
Minimal Impact	No measurable latency or throughput degradation introduced by CNFW enforcement; CPU/memory overhead stable under enterprise real-world traffic mixes.	**100%**
Moderate Impact	Slight latency or overhead under peak load conditions but acceptable for enterprise cloud operations.	**75%**

High Impact but Functional	Noticeable throughput reduction or resource spikes under load; east-west inspection introduces latency but firewall remains operational.	**25%**
Unacceptable Performance	Severe performance degradation renders cloud workload protection operationally unreliable.	**0%**

Incident Response and Visibility:

Outcome	Description	Score
Proactive & Granular	Real-time alerts with rich workload-context metadata; immutable cloud audit trails; forensic event reconstruction; intuitive console for rapid incident response.	**100%**
Effective but Limited	Alerts and logs are reliable but lack deep forensic context (workload identity, IAM role) or advanced console investigation capabilities.	**75%**
Reactive & Manual	Incidents visible only via manual log analysis in cloud-native logging pipelines; slow response workflows; limited alert integration.	**25%**
Blind & Ineffective	No useful security logs or alerts; cloud-native incidents undetectable in real time.	**0%**

Security Administration:

Outcome	Description	Score
Granular & Auditable	Strong RBAC across cloud accounts; tamper-proof compliance logs; frequent automated threat intelligence patching; alerts integrated with enterprise incident response platforms.	**100%**
Secure but Limited Audit	RBAC enforced; threat intelligence updates timely; audit and alerting integration limited to basic SIEM forwarding.	**75%**

Weakly Controlled	Basic RBAC; irregular signature updates; logs incomplete; alerts unreliable or missing cloud-context metadata.	**25%**
Uncontrolled & Non-Compliant	No meaningful RBAC; irregular or manual patching; no compliance logs; weak or absent alerting capabilities.	**0%**

6.3 COMPLIANCE VALIDATION SCORING CRITERIA

The scoring criteria offer a structured framework to quantify the compliance effectiveness of the Cloud-Native Firewall. They assess regulatory alignment, audit readiness, data protection enforcement, and governance across cloud environments. Each area is evaluated using a four-tier scale.

Outcome	Description	Score
Compliant & evidenced	Policies and controls fully aligned with regulatory requirements; events logged comprehensively; immutable audit trails generated automatically; compliance reports exportable.	**100%**
Compliant, No Evidence	Controls meet core regulatory requirements but logging, reporting, or automated audit trail generation is incomplete.	**75%**
Non-Compliant but Evidenced	Some regulatory controls implemented but significant gaps exist; visibility and compliance reporting limited.	**25%**
Non-Compliant	No prevention, no logging, no compliance support. Controls absent or non-functional.	**0%**

7. GENERAL EVALUATION APPROACH

7.1 CLOUD-NATIVE FIREWALL VENDOR PARTICIPATION SELECTION CRITERIA

SecureIQLab will select Cloud-Native Firewall vendors for participation in the validation study based on the following criteria:

A. Market Leaders

- Vendors recognized as leaders in terms of global cloud security revenue, customer adoption across major cloud platforms, or strong channel presence.

B. Analyst-Recognized and Enterprise Challengers

- Vendors identified through industry analyst reports (e.g., Gartner Magic Quadrant, Forrester Wave, IDC MarketScape, Buyer's Guides) covering cloud network security.
- Vendors referenced by enterprise cloud security professionals through surveys, direct inquiries, and feedback from enterprises, organizations, MSPs, and MSSPs.

C. Innovative New Entrants

- Emerging vendors with breakthrough cloud-native firewall technology offerings including container-native, serverless, or cloud-mesh architectures.
- Vendors demonstrating interest in independent validation to establish market credibility.

Conflict of Interest Statement: SecureIQLab affirms that there are no known conflicts of interest in the vendor selection process. Vendor inclusion is based solely on transparent, objective, and market-relevant criteria.

7.2 SCOPE

For this iteration of the SecureIQLab Cloud-Native Firewall Cyber Risk Validation, the evaluation focuses exclusively on CNFW solutions that are available in cloud marketplaces or delivered as cloud-ready/SaaS-based deployments. Examples of qualifying cloud marketplaces include AWS Marketplace, Azure Marketplace, Google Cloud Marketplace, SaaS, and FWaaS offerings. Vendors may choose to include different products for different cloud platform offerings.

The following vendors and products have been selected for inclusion in this validation:

Vendor	Product Name
Palo Alto Networks	CN-Series Containerized Firewall
Check Point Software Technologies	CloudGuard Network Security
Fortinet	FortiGate CNF (Cloud-Native Firewall)
Cisco	Cisco Secure Firewall Cloud Native (SFCN)
Zscaler	Zscaler Cloud Firewall (ZIA)
Barracuda Networks	CloudGen Firewall
Google Cloud	Cloud NGFW Enterprise
Cloudflare	Cloudflare Magic Firewall
Microsoft	Azure Firewall Premium
Amazon Web Services	AWS Network Firewall
Netskope	Netskope One (Next-Gen Firewall)
Aviatrix	Cloud Native Security Fabric
Juniper Networks	vSRX Virtual Firewall
Cato Networks	Cato FWaaS
Illumio	Illumio CloudSecure
NordLayer	NordLayer Cloud Firewall

7.3 VALIDATION TIMELINE

The SecureIQLab Cloud-Native Firewall Cyber Risk Validation will be executed in five phases. The test plan remains within scope if the project remains within four weeks of the timeline below.

Index	Test Activity	Date Range	Dependencies
1	Test Commencement	6 Oct 2026	Vendor voluntary participation (or) procurement of vendor software.
2	Confirm Vendor Configuration Feedback	13 Oct – 30 Oct 2026	All required vendors installed, smoke tested, and configurations validated by vendors where possible.
3	Testing	30 Oct – 19 Dec 2026	Based on smoke test results, disputes, and resolution.
4	Feedback and Dispute Resolution Retests as Needed	19 Dec 2026 – 10 Jan 2027	Based on report feedback and final dispute resolution.
5	Publish Results	15 March 2027	Dependent on vulnerability disclosure requirements.

Note: Dates may be adjusted to accommodate vendor availability, remediation activities, or disclosure considerations.

7.4 RISK AND MANAGEMENT

No additional risks are known at this time.

7.5 GEO LIMITATION

While performing cloud-centric attack simulations, SSRF exploits, and post-exploitation test cases, SecureIQLab will make every effort to use only attacks that are not geo-location centric when necessary. SecureIQLab will ensure that attacks originate from as wide a range of IP addresses and cloud regions as possible.

7.6 DISTRIBUTION OF TEST DATA

Upon the completion of the validation project phases, the resulting data will be organized into individual test reports and one comparative report. These results will be available for vendors to purchase for marketing and will also be publicly available to download at https://secureiqlab.com/publications/

7.7 FUND AGREEMENT

This is a non-commissioned test funded by SecureIQLab.

7.8 DISPUTE PROCESS

SecureIQLab will make best efforts to resolve disputes regarding scoring. Any changes to scoring resulting from successful disputes will be applied to all vendor results, and not just to the disputing vendor.

All Cloud-Native Firewall vendors who participate in this test will receive their score. This will include a breakdown of security efficacy, operational efficacy, and compliance efficacy scores. This data set will be shared individually with the Cloud-Native Firewall vendors and SecureIQLab will work closely to review the metrics as well as relevant metadata where warranted. Furthermore, SecureIQLab will not share attacks that are missed during the testing window to third parties unless required by law. SecureIQLab will provide vendors with up to two weeks for the dispute resolution on the nature of attacks. Any security vulnerabilities uncovered during the testing windows related to the Cloud-Native Firewall under test will be shared based upon responsible disclosure policy, giving vendors up to 20 days to remediate the vulnerability. Vulnerability details will be disclosed to the broader public when a fix is available, or when it is in the interest of the public.

SecureIQLab will not entertain disputes or changes to scoring after the Comparative and Individual Test reports have been published.

7.9 OPT-OUT-POLICY

Opt-Out: Opt-out will only be considered for the following reasons:

- The product, solution, or technology is found to be outside of scope in the context of the methodology as determined by SecureIQLab.
- Any technology, product, or solution that is NOT generally available nor ready for deployment.
- Publishing the test would not serve the public interest as deemed by SecureIQLab.

Opt-out requests must be provided in writing. Emailed opt-outs must be sent to info@secureiqlab.com. Mailed opt-outs must be sent to:

SecureIQLab

9600 Great Hills Trail Suite 150W

Austin, TX 78759

Mailed opt-outs are effective by the date received, not the date posted. We do not accept opt-outs through phone, voice, social media, or similar channels. The opt-out must contain the name, title, email, and phone number of the individual authorized to request an opt-out on behalf of the vendor. To be considered a completed opt-out, the request must state under which of the reasons above the request should be considered and provide details to support the request. All vendors have a limited right to opt-out for the designated reasons listed above. The opt-out period begins at Test Commencement and continues through the end of the Dispute Phase [Section 4.6]. Vendors will be contacted by SecureIQLab within 3 business days of receiving the opt-out request to discuss feasibility. If a vendor opts out before the end of the Configuration Phase, the vendor will be listed as 'Participant, not tested'. If a vendor opts out after testing has been performed for their product, their product will be marked in the results as 'Tested, not published'.

8. ATTESTATION

I understand and agree that I am submitting this Test Plan, and the following Attestations, on behalf of the entity listed below, and I represent and warrant that I have authority to bind such entities to these Attestations. All references to "I" or "me" or similar language refer to such an entity. I represent and warrant that the following Attestations are true, to the best of my knowledge and belief, and each of the following commitments will be upheld to the best of my ability.

I will provide public notification on the AMTSO website covering my obligation for notification of a Public Test, regardless of whether a potential Participant is in actual receipt of such notification prior to the Commencement Date of a Test.

All products included in this Test will be analyzed fairly and equally.

I will disclose any anticipated or known imbalance or inequity in the Test design to all Participants in the Test.

Although I may charge for participation in a Test, I will not charge any additional fees for a vendor to be a test subject under the Standards.

I will disclose any material conflicts of interest or other information that could materially impact the reliability of the Test.

I will disclose how the Test was funded.

I hereby affirm, to the best of my knowledge and belief that this Test Plan complies with the AMTSO Testing Standards, as of the date hereof.

Signature: /s/ David Ellis

Name: David Ellis

Test Lab: SecureIQLab

AMTSO Test ID: XXXXXXXXXXXXX

9. DOCUMENT VERSION

Version	Section	Revision Overview
V 1.0	1	Restructured Introduction to align with SecureIQLab Enterprise Cyber Risk Validation format; added Cloud-Native vs. Traditional Firewall Distinctions; added Cloud Threat Landscape & Emerging Risks; added Enterprise Adoption Drivers; added Rationale for Independent Cyber Risk Validation.
V 1.0	2.2	Added four structured objective pillars: Security Efficacy, Operational Efficacy, Compliance Efficacy, and Vendor-Neutral Comparisons.
V 1.0	3.1.1	Restructured Threat Defense into: Application-Based Threats, Vulnerability-Based Attacks, Malware & Botnet Defense, Browser-Based Threats, Data Loss & Cloud Storage Leakage, Container & Serverless Security, GenAI Workload Security.
V 1.0	3.1.2	Restructured Policy Enforcement to include Identity-Aware Policies and MCP Server controls alongside existing Stateful Inspection, Application Control, Geo Location, Service Control, Web/URL Filtering, IP/Port, and TOR Exit Node categories.
V 1.0	3.1.3	Created Advanced Evasive Techniques section incorporating protocol tunneling, polymorphic payloads, timing-based evasion, APTs, and cloud-centric post-exploitation techniques.
V 1.0	3.1.4	Created Encryption Capabilities section covering SSL/TLS cipher support, TLS session reuse, and SSL scaling for cloud-native environments.
V 1.0	3.2	Operational Efficacy restructured and expanded with cloud-native deployment, IaC policy management, ecosystem integration, scalability, performance resiliency, incident response, reporting, and security administration categories.
V 1.0	3.3	Compliance Efficacy introduced covering GDPR, HIPAA, PCI DSS, NIST 800-171, SOC 2, ISO/IEC 27001:2022, and Secure by Design/Default principles.
V 1.0	6	Scoring criteria framework introduced with three dimensions: Security Efficacy (Prevention + Detection), Operational Efficacy (seven sub-categories), and Compliance Efficacy.

10. COPYRIGHT AND DISCLAIMER

For more information about SecureIQLab and the testing methodologies, please visit our website.

SecureIQLab (March 2026)