NIST SECURITY STANDARDS FOR MICROSERVICES

SP 800-204 SERIES REQUIREMENTS GUIDE

tetrate

OVERVIEW

NIST SP 800-204 SERIES

REQUIREMENTS GUIDE

NIST's 800-204 series of Special Publications provides a comprehensive set of security strategies for microservices applications and is part of a critical set of documents defining the US government's standards of zero trust. Co-authored with NIST by Tetrate founding engineer Zack Butcher, the series focuses mainly on strategies for implementing a zero trust network architecture and secure communications between microservices as well as between microservices and external systems and end users.

WHO SHOULD READ THIS?

This document offers a concise representation of NIST's specific recommendations paired with notes on how Tetrate's flagship application connectivity platform, Tetrate Service Bridge, implements the standard.

This document is for **security professionals**, **platform owners**, and **application developers** looking for a compilation of NIST's security standards for microservices-based applications.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

LEGEND

IMPLEMENTATION OF NIST RECOMMENDATIONS

Support for each requirement is described as follows:

TSB	*Solution available out of the box with Tetrate Service Bridge, including built-in support for multi-cluster, multi-cloud, and hybrid-cloud deployments and heterogeneous compute environments.*
Istio	*Possible with Istio and Envoy.*
Kubernetes	*Provided by the underlying compute platform (e.g., Kubernetes).*

SECTION 1

NIST SP 800-204

SECURITY STRATEGIES FOR MICROSERVICES-BASED APPLICATION SYSTEMS

SP 800-204, the first paper in the series, provides context on the technology and threat background for microservices applications. It goes on to outline a core set of features that must be available in the infrastructure environment of a microservices environment to properly address the unique security and availability concerns of a microservices architecture.

MS-SS-1: Security Strategies for Authentication

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
Authentication to microservices APIs that have access to sensitive data should not be done simply by using API keys. Access to such APIs should require authentication tokens that have either been digitally signed (e.g., client credentials grant) or is verified with an authoritative source. Additionally, some services may require either single-use tokens or short-lived tokens (tokens that expire after a short time period) to limit the damage a compromised token can cause.	*Istio supports request-level, end-user authentication with JSON Web Token (JWT) validation using either a custom authentication provider or any OpenID Connect (OIDC) provider. Tetrate Service Bridge enables centralized authorship and global enforcement of token lifecycle policy.*	TSB Istio
Authentication tokens should be handle-based (where initially a token reference is sent to Relying party (RP)), cryptographically signed, or protected by an Hash-based Method Authentication Code (HMAC) scheme.	*See above. JWTs satisfy the requirements of this point.*	TSB Istio
Every API Key that is used in the application should have restrictions specified both for the applications (e.g., mobile app, IP address) and the set of APIs where they can be used.	*Istio supports JWTs, which provide for scope claims that can be used to author policy for enforcing specific audiences.*	TSB Istio

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
The restriction scope for functionality of every API Key should be commensurate with the level of assurance provided during identity proofing, whether it be machine or human driven identity proofing.	*This is the responsibility of the authorization system that mints the JWT.*	N/A
When stateless authentication tokens (e.g., JSON Web Tokens (JWT)) are used by implementing shared libraries associated with a microservice, the following security precautions must be observed: (a) the token expiry times should be as short as possible since they determine the duration of the session and an active session cannot be revoked, and (b) the token secret key must not be a part of the library code; it must be a dynamic variable represented by an environmental variable or specified in an environment data file. The key value should be stored in a data vault solution.	*Istio supports short token expiry times. As a dedicated infrastructure layer that acts as a security kernel, Istio provides assurance of best practices, minimizing accidental hazards like storing secrets in source repositories and library code. It uses a custom Kubernetes API to securely store policy configuration and keys. TSB ensures that policies like token expiry times are enforced consistently across environments.*	TSB Istio
If standards-based techniques such as OAuth or OpenID connect are implemented, they must be deployed securely.	*While deployment of OAuth or OIDC is outside the scope of Tetrate Service Bridge, Tetrate can advise on security best practices. Hosting authentication and authorization services in the mesh is a core technique for deploying authentication services securely. Also, TSB has built-in support for external authentication and authorization services.*	N/A

MS-SS-2: Security Strategies for Access Management

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
Access policies to all APIs and their resources should be defined and provisioned to an access server. Access policies at a coarse level of granularity say "Permit to Call for a given set of addressable functionalities" should be defined and enforced at the initial API gateway while authorizations at the finer level of granularity (e.g., related to domain of the particular microservices' business logic) should be defined and enforced closer to the location of the microservices (e.g., at the micro gateway) or sometimes at the microservice itself.	*Istio provides two levels of authentication: 1) peer-to-peer authentication between services using SPIFFE and mTLS; and 2) request authentication used for end-user authentication to verify credentials attached to requests. Istio supports local authorization using JWT claims and external authorization (ext-authz), which uses a service running externally to render a verdict.* *TSB facilitates centralized policy authorship plus configuration and management of multiple Istio control planes to ensure authorization policy is consistently implemented everywhere.`*	TSB Istio
Caching Mechanism: It may be appropriate to allow microservices to cache policy data; this cache should be only relied upon when an access server is unavailable and should expire after a duration appropriate for the environment/infrastructure.	*Istio doesn't directly implement caching of policy data, but does provide infrastructure for it by way of supporting, for example, attaching cache keys to request headers.*	N/A
The access server should be capable of supporting fine-grained policies.	*Istio supports fine-grained policy for service-to-service access control, incorporating end-user credentials; however, end-user to resource access control is out of scope.*	N/A

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
Access decisions from the access server should be conveyed to individual and sets of microservices through standardized tokens encoded in a platform-neutral format (e.g., OAuth 2.0 token encoded in JSON format). The token can be either a handle-based token or an assertion bearing token.	*While Istio doesn't mint tokens, it supports access control via JWT scope codes, for example.*	N/A
The scope of internal authorization tokens appended by the micro gateway or decision point to each request should be carefully controlled; for example, in a request for transaction, the internal authorization token should be limited in scope to only involve the API endpoints that must be accessed for that transaction.	*The best practice here is to re-mint a token at every step that strips away unnecessary privileges. While Istio doesn't support minting new tokens, it offers hooks to do so with your own server.*	N/A
The API gateway can be leveraged to centralize enforcement of authentication and access control for all downstream microservices, eliminating the need to provide authentication and access control for each of the individual services. If this design is chosen, any component suitably positioned on the network can make anonymous connections to the services bypassing the API gateway and its protections. Mitigating controls such as mutual authentication should be leveraged to prevent direct, anonymous connections to the services.	*TSB offers the best of both worlds: API gateway capabilities available from edge to workload plus the security kernel guarantees of a service mesh. Enforcement of authentication and access control is available at the gateway and workload and may be applied where most appropriate. In addition, all communication between services and with the outside world may be secured with mutual authentication and encryption.*	TSB Istio

MS-SS-3: Service Registry Configuration

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
Service registry capabilities should be provided through servers that are either dedicated or part of a service mesh architecture.	*Istio's control plane maintains an internal service registry used to configure Envoy to route requests to the relevant services. Istio automatically detects the services and endpoints in a Kubernetes cluster. TSB's global control plane can augment the Kubernetes-native service discovery mechanism to populate the Istio service registry in each cluster with entries for cluster-external services, such as those running in other availability zones, clouds, or in non-Kubernetes workloads (such as virtual machines and bare metal servers). TSB's global control plane ensures the consistency of each Istio service registry instance across clusters.*	TSB Istio Kubernetes
Service registry services should be in a network that has been configured with certain Quality of Service (QoS) parameters to ensure its availability and resilience.	*Network quality of service between Istio's data plane and the service registry in the control plane is provided by Kubernetes.*	Kubernetes
Communication between an application service and a service registry should occur through a secure communication protocol such as HTTPS or Transport Layer Security (TLS).	*Each Envoy instance in the data plane communicates with the service registry in the Istio control plane via TLS.*	TSB Istio

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
Service registry should have validation checks to ensure that only legitimate services are performing the registration, refresh operations, and database queries to discover services.	*Service registry validation is performed by the Kubernetes platform.*	Kubernetes
The bounded context and loose coupling principle for microservices should be observed for the service registration/deregistration functions. In other words, the application service should not have tight coupling with an infrastructure service, such as a service registry service, and service self-registration/deregistration patterns should be avoided. When an application service crashes or is running but unable to handle requests, its inability to perform deregistration affects the integrity of the whole process. Therefore, registration/ deregistration of an application service should be enabled using a third-party registration pattern, and the application service should be restricted to querying the service registry for service location information as described under the client-side discovery pattern.	*Service registry functions are managed in a dedicated infrastructure layer outside the scope of application code. Microservices do not self-register or deregister and, in fact, application code cannot service registration functions. Istio integrates with the Kubernetes platform to do service discovery; Kubernetes handles registration and deregistration automatically.*	TSB Istio Kubernetes
If a third-party registration pattern is implemented, registration/deregistration should only take place after a health check on the application service is performed.	*Istio works with Kubernetes liveness and readiness probes to ensure service health. When virtual machines (outside of Kubernetes, by definition) are enrolled into the service mesh, Istio performs liveliness and readiness checks on them as well.*	TSB Istio Kubernetes

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
Distributed service registry should be deployed for large microservices applications, and care should be taken to maintain data consistency among multiple service registry instances.	*TSB's global control plane ensures the consistency of each Istio service registry instance across clusters.*	TSB Istio

MS-SS-4: Secure Communication

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
Clients should not be configured to call target services directly but rather to point to the single gateway URL.	*Istio ingress gateways present a single gateway URL to external clients, performing client-side load balancing automatically and routing client requests to appropriate internal services.*	TSB Istio
Client to API gateway as well as Service to Service communication should take place after mutual authentication and be encrypted (e.g., using mutual TLS (mTLS) protocol).	*All communication in the mesh can be configured to require mTLS.*	TSB Istio
Frequently interacting services should create keep-alive TLS connections.	*Connection keepalive and connection upgrade from HTTP/1.1 to HTTP/2 is part of Istio's suite of resiliency capabilities including load balancing, timeout, circuit breaker, etc. Better than just keepalive, connection upgrade to HTTP/2 allows services to multiplex requests in parallel on the same connection, reducing the total number of connections and avoiding head-of-line blocking.*	TSB Istio

MS-SS-5: Security Monitoring

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
Security monitoring should be performed at both the gateway and service level to detect, alert and respond to inappropriate behavior, for example a bearer token reuse attack and injection attacks. Further, input validation errors and extra parameters errors, crashes and core dumps must be logged. A class of software that can accomplish this is the Open Web Application Service Project (OWASP) AppSensor which could be potentially implemented in the gateway, service mesh and microservice itself.	*With Envoy as the consistent data plane, TSB offers capabilities traditionally limited to the edge or DMZ anywhere in our application traffic platform. Envoy produces a rich set of telemetry data that can be fed into any monitoring system.* *TSB combines a range of Envoy's features together into an easy-to-use package to enable API gateway features like token validation, rate limiting, and OpenAPI spec-based configuration. It also brings ModSecurity-based WAF capabilities to sidecars, ingress gateways, and edge load balancers. Best of all, TSB supports writing a single policy and applying it to traffic anywhere: between external clients and your services, across clusters or data centers in your network, or even between services running on the same cluster.t*	TSB Istio
A central dashboard displays the status of various services and the network segments that link them. At a minimum, the dashboard should show security parameters such as input validation failures and unexpected parameters that are obvious signs of injection attack attempts.	*Kiali offers a dashboard for services in a single cluster; TSB offers a global dashboard showing the topology and connections between services.*	TSB Kiali

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
A baseline for normal, uncompromised behavior in terms of the outcome of business logic decisions, contact attempts, and other behavior should be created. The placement and capabilities of Intrusion Detection System (IDS) nodes should be such that deviations from this baseline can be detected.	*Istio can feed data to an existing intrusion detection system and enforce verdicts, but does not, by itself, play the role of an IDS.*	TSB Istio External IDS

MS-SS-6: Circuit Breakers

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
A proxy circuit breaker option should be deployed to limit the trusted component to the proxy. This avoids the need to place the trust on the clients and microservices (e.g., setting thresholds and cutting off requests based on the set threshold) since they are multiple components.	*Resiliency features, in their capacity to ensure continuity of service, are also security features. Circuit breaking helps prevent cascading failures and is available as part of Istio's out-of-the-box resiliency capabilities. These resiliency capabilities help bound the behavior of the system. With a narrower range of behaviors, the system itself becomes more understandable, easier to manage, and more amenable to monitoring against a baseline.*	TSB Istio

MS-SS-7: Load Balancing

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
All programs supporting the load balancing function should be decoupled from individual service requests. For example, the program that performs health checks on services to determine the load balancing pool should run asynchronously in the background.	*Istio performs client-side load balancing with health checks. The health checks are coupled to individual service requests to avoid the chattiness of dedicated polling, but Istio also implements automatic retries to hide it from the application.*	TSB Istio
Care must be taken to protect the network connection between the load balancer and the microservice platform.	*All communication in the mesh can be configured to use mTLS. TSB's hierarchical policy allows security teams to transparently mandate and audit secure communication between application components without burdening application teams with security implementation concerns.*	TSB Istio
When a DNS resolver is deployed in front of a source microservice to provide a table of available target microservice instances, it should work in tandem with the health check program to present a single list to the calling microservice.	*Rather than use DNS, Istio's control plane integrates with the Kubernetes platform to perform service discovery and updates exact endpoints in real time, which is faster and better than a DNS resolver.*	TSB Istio Kubernetes

MS-SS-8: Rate Limiting

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
Quotas or limits for application usage should be based on both infrastructure and application-related requirements.	*TSB supports rate limiting within a cluster or across multiple clusters, using either a built-in or external rate limit server, although specific policy configurations will depend on the application. Developing a plan for quotas or limits is out of scope; once established, TSB will enforce usage limit policy.*	N/A
Limits should be determined based on well-defined API usage plans.	*See above.*	N/A
For high security microservices, replay detection must be implemented. Based on the risk, this feature can be configured to detect replays 100% of the time or perform random detection.	*While TSB doesn't implement replay detection itself, encryption-in-transit via TLS prevents most replay attacks and JWTs protect against stolen credentials.*	N/A

MS-SS-9: Service Version Updates

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
The traffic to both the existing version and the new version of the service should be routed through a central node, such as an API gateway, to monitor that the blue/green transition occurs in a controlled manner and to monitor the risk associated with a canary release. Security monitoring should cover nodes hosting both the existing and newer versions.	*Istio has built-in support for blue/green and canary releases via client-side load balancing. As such, traffic doesn't have to go through a central node to implement routing policy. Tetrate Service Bridge supports blue-green and canary deployments across clusters via Tier 1 gateways.*	TSB Istio
Usage monitoring of the existing version should steadily increase traffic to the new version.	*Istio provides consistent operational metrics across every application in a cluster. Tetrate Service Bridge consolidates metrics from every cluster into a single view, providing a global view of application behavior that enables intelligent runtime decisions about when and where to route traffic.*	N/A
The performance and functional correctness of the new version should be factors in increasing traffic to the new version.	*Istio doesn't automatically detect performance and functional correctness; however, signals from the mesh can be used to make such a decision, especially in conjunction with delivery tools like Flagger and Argo CD.*	N/A
Client preference for the version (existing or new) should be taken into consideration while designing a canary release technique.	*Istio offers fine-grained traffic shaping, including routing based on client headers and similar to support this behavior.*	TSB Istio

MS-SS-10: Persistent Session Integrity

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
The session information for a client must be stored securely.	*Istio does not store session information. What client information TSB does store, e.g., credentials, is kept encrypted in the browser.*	N/A
The artifact used for conveying the binding server information must be protected.	*See above.*	N/A
Internal authorization tokens must not be provided back to the user, and the user's session tokens must not be passed beyond the gateway for use in policy decisions.	*While out of scope of the service mesh itself, Istio supports such credential exchange by, e.g., interacting with the authorization server at the gateway to swap external credentials for internal credentials.*	N/A

MS-SS-11: Credential Abuse and Stuffing Attacks

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
A run-time prevention strategy for credential abuse is preferable to an offline strategy. A threshold for a designated time interval from a given location (e.g., IP address) for the number of login attempts should be established; if the threshold is exceeded, preventive measures must be triggered by the authentication/ authorization server. This feature must be present when a bearer token is used, to detect its reuse and enforce prevention.	*Credential abuse detection and prevention is the purview of the underlying identity, credential, and access management (ICAM) system, but Istio can apply mitigation efforts such as rate limiting per IP as needed.*	N/A
A credential-stuffing detection solution has the capability to check user logins against the stolen credential database and warn legitimate users that their credentials have been stolen.	*See above.*	N/A
Configure IDS and boundary devices to detect the following: (a) a denial of service attack and raise an alert before the service is no longer accessible, and (b) a distributed network probe.	*See above.*	N/A
Configure service hosts to scan file uploads and the contents of each container's memory and file system for resident malware threats.	*See above.*	N/A

MS-SS-12: API Gateway Implementation

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
Integrate the API gateway with an identity management application to provision credentials before activating the API.	*As a founding member of the Envoy Gateway steering committee, Tetrate is committed to supporting API gateway capabilities inside Envoy. As such, TSB provides API gateway capabilities from application edge to workload. TSB supports end-user authentication via OIDC and can enforce authentication policy at any Envoy instance—from Tier 1 gateways at the application edge to ingress gateways at the cluster edge, and at individual workloads.*	TSB
When identity management is invoked through the API gateway, connectors should be provided for integrating with identity providers (IdPs).	*TSB supports integration with a variety of identity providers via LDAP, OIDC, and JWT.*	TSB
The API gateway should have a connector to an artifact that can generate an access token for the client request (e.g., OAuth 2.0 Authorization Server).	*Istio supports such a connector subject to configuration by platform operations.*	TSB Istio
Securely channel all traffic information to a monitoring and/or analytics application for detecting attacks (e.g., denial of service, malicious actions) and unearthing explanations for degrading performance.	*TSB uses SkyWalking to collect metrics and relay them to monitoring and analytics applications—including the dashboards built into TSB—via an authenticated and encrypted TLS channel. Attack detection based on that data is beyond the scope of TSB.*	TSB

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
Distributed gateway deployments (or a combination of initial gateway (that intercepts all client accesses) and microgateways (closer to microservices)) should have a token translation (exchange) service [18] between gateways. The token presented to the initial gateway should have permissions with a broad scope whereas the token presented to inside gateways (or microgateways) should be more narrowly scoped with specific permissions or an entirely different token type that is appropriate for the target microservice platform. This helps to implement the least privilege paradigm.	*Similar to MS-SS-2 above, Istio provides hooks necessary to integrate with an underlying authentication system to implement this behavior.*	TSB Istio External Auth System

MS-SS-13: Service Mesh Implementation

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
Provide policy support for designating a specific communication protocol between pairs of services and specifying the traffic load between pairs of services based on application requirements.	*Istio offers a rich set of traffic management capabilities including routing, protocol translation, and resilience features like timeouts, retries, circuit breakers, and fault injection. Tetrate Service Bridge extends those capabilities beyond a single cluster to include traffic management across clusters, clouds, and data centers.*	TSB Istio
The default configuration should always enable access control policies for all services.	*As a security kernel for microservices, Tetrate Service Bridge provides centrally managed, globally enforced access control policies by default.*	TSB Istio
Avoid configurations that may lead to privilege escalation (e.g., the service role permissions and binding of the service role to service user accounts).	*The configuration and deployment of application containers and other application workloads is managed by the underlying compute platform and, as such, is outside the scope of TSB. TSB components themselves do not run as privileged containers.*	Kubernetes
Service mesh deployments should have configuration capabilities to specify resource usage limits for its components. The absence of this feature creates the potential for these components to impact the resiliency and availability of the overall microservices application.	*Istio supports configuration for resource usage limits. TSB exposes them for multi-cluster configuration.*	TSB Istio

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
Service mesh deployments should have configuration capabilities to collect and send environment metrics, including request metrics, to a centralized service for monitoring. Policies should allow for specifying either a single service mesh or multiple service meshes (each with their own control plane) for multi-cluster microservices environments to ensure high availability and resiliency in those scenarios.	*Istio provides consistent operational metrics across every application in a cluster. Tetrate Service Bridge consolidates the metrics from every cluster enrolled in the mesh into a single view, providing a global view of application behavior that enables intelligent decisions about when and where to route traffic.*	TSB Istio
For highly sensitive microservices-based applications, Layer 3 network segmentation must be configured within the orchestrator platform to complement the Layer 5 network segmentation achieved throughout the service mesh layer. This is a countermeasure to the threat by malicious actors circumventing or bypassing the sidecar proxy that the service mesh uses for firewalling and blocking network traffic.	*Defense-in-depth and aligning CNI with service mesh is a best practice, but service mesh operates at Layer 7, not Layer 5.*	N/A

SECTION 2

NIST SP 800-204A

BUILDING SECURE MICROSERVICES-BASED APPLICATIONS USING SERVICE MESH ARCHITECTURE

SP 800-204A, the second paper in the series, establishes a reference platform for delivering microservices security, consisting of Kubernetes as the orchestrator and the Istio service mesh as the security kernel. Establishing this reference platform enables NIST to demonstrate specific implementations of the concepts presented. The paper then provides a series of recommendations for implementing the core features presented in the first paper, SP 800-204.

4.1: Communication Configuration for Service Proxies

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SM-DR1: ALLOWED TRAFFIC There should be a feature to specify the set of protocols and ports into which a service proxy can accept traffic for its associated service. By default, a service proxy should not allow traffic except as specified by this configuration.	*TSB requires specific protocols and ports for inbound traffic to be explicitly allowed based on the Kubernetes Service definition or the Istio service entry. If not explicitly allowed, inbound traffic is blocked.*	TSB Istio
SM-DR2: REACHABILITY The set of services that a service proxy can reach must be limited. There should be features to limit access based on namespace, a specific named service within a given namespace, or the service's runtime identity. Access to the control plane of the Service Mesh must always be provided to relay discovery, different policies, and telemetry data.	*TSB can be configured to require outbound traffic from each service to be explicitly allowed. If not explicitly allowed, outbound traffic to other services—both internal and external to the mesh—is blocked by the sidecar or the egress gateway. The Envoy instances in the data plane maintain connections to the Istio control plane for configuration updates and to relay telemetry data.*	TSB Istio

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SM-DR3: PROTOCOL TRANSLATION The service proxy should have built-in capabilities to support clients communicating with different protocols than the target microservice (e.g., convert REST/HTTP requests to gRPC requests or upgrade HTTP/1.1 to HTTP/2). This is required to avoid the need for building a separate server per client protocol, which increases the attack surface.	*Istio offers a rich set of traffic management capabilities including routing, protocol translation, and resilience features like timeouts, retries, circuit breakers, and fault injection. Tetrate Service Bridge provides monitoring of these capabilities and extends them beyond a single cluster to include traffic management across clusters, clouds, and data centers.*	TSB Istio
SM-DR4: USER EXTENSIBILITY The service proxy should have features for defining custom logic in addition to built-in logic for handling network functions. This is required to ensure that the service proxy can be extended to implement use case specific policies (e.g., preexisting or home-grown policy engines). The implementation should provide means to control the risks of extensibility, such as sandboxing, restricting the APIs/runtimes of languages used, or performing pre-analysis that ensures safety (e.g. WASM or eBPF).	*The Envoy-based data plane is highly extensible via native C++ filters as well as Wasm plugins, or by implementing Envoy APIs out-of-process. Wasm extensions are isolated from the host environment and executed in a memory-safe, sandboxed virtual machine.*	TSB Istio

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SM-DR5: DYNAMIC CONFIGURATION There should be options to configure proxies dynamically (e.g., event-driven configuration updates) in addition to static configuration. In other words, there should be discovery services for those entities that are expected to be dynamic rather than known at the time of deployment. Further, the proxy should atomically swap to the new dynamic configuration at runtime while efficiently handling (i.e., completing or terminating) outstanding requests under the previous configuration. This is required for timely enforcement of policy changes at runtime without any degradation of user traffic or downtime (i.e., without restarting the service proxy).	*Envoy supports dynamic (runtime) configuration, connection draining, and hot restart. Envoy also supports swapping new dynamic configuration atomically at runtime. Connections are cycled regularly—on the order of hours, not days or weeks—and not just on configuration updates. This ensures that connection policy, including mutual authentication during connection handshake, is applied regularly.*	TSB Istio
SM-DR6: SERVICE TO PROXY COMMUNICATION CONFIGURATION The application service and its associated proxy should only communicate through a loopback channel (e.g., localhost IP address, UNIX domain socket, etc.). Further, service proxies should only communicate with each other by setting up a mutual TLS (mTLS) session where every exchanged data packet is encrypted.	*Istio uses iptables to intercept and mediate traffic between service instances and the network, ensuring that, as a security kernel, it is non-bypassable. It can also be configured to use domain sockets to bypass the network stack altogether. In addition, Istio provides strong identity for each service instance and ensures mTLS communication links between services. Tetrate Service Bridge facilitates globally-consistent security policy by ensuring policy configuration and enforcement across multiple clusters, clouds, and data centers.*	TSB Istio

4.2: Configuration for Ingress Proxies

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SM-DR7: INGRESS PROXIES There should be features for configuring traffic routing rules for ingress (standalone) proxies just like service proxies. This is needed because consistent enforcement of routing policy is required all the way to the edge of the application deployment.	*Istio offers consistent traffic management capabilities and configuration at cluster ingress and sidecars. Tetrate Service Bridge extends mesh traffic management from edge to workload to provide consistent ingress controls across clusters, clouds, and data centers.*	TSB Istio

4.3: Access to External Services

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SM-DR8: RESTRICTING ACCESS TO EXTERNAL RESOURCES Access to external resources or services outside of the mesh should be disabled by default and only allowed by an explicit policy that restricts access to specified destinations. Additionally, those external resources or services should be modeled as services in the Service Mesh itself (e.g., by including them in the Service Mesh's service discovery mechanism).	*Istio's Envoy-based egress gateway can enforce policy on connections from services to external resources. Tetrate Service Bridge maintains a global service registry that allows enterprises to model all resources in the mesh—including external services—that may be used by applications across an entire fleet. Out of the box, Istio adopts a permissive policy allowing connections to any external resource so as not to break existing applications. However, as a best practice, we recommend configuring the mesh to allow access only to the services in the registry.*	TSB Istio
SM-DR9: SECURE ACCESS TO EXTERNAL RESOURCES The same availability improvement features (e.g., retries, timeouts, circuit breakers, etc.) that are configured for services inside of the Service Mesh must be provided for access to external resources and services.	*For the applications in the mesh, all of the traffic management and resiliency capabilities available for inbound traffic to internal resources and between services are also available for outbound traffic to external resources. This can be done either via a dedicated egress proxy or at individual sidecars.*	TSB Istio

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SM-DR10: EGRESS PROXIES Access to external resources or services outside of the mesh should be disabled by default and only allowed by an explicit policy that restricts access to specified destinations. Additionally, those external resources or services should be modeled as services in the Service Mesh itself (e.g., by including them in the Service Mesh's service discovery mechanism).	*Like Istio's Envoy-based ingress gateway, Istio's egress gateway mediates connections to external resources to implement these controls (see SM-DR7 and SM-DR8 above).*	TSB Istio

4.4: Configuration for Identity and Access Management

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SM-DR11: UNIVERSAL IDENTITY DOMAIN The identity of all instances of a microservice should be consistent and unique—consistent in that a service should have the same name regardless of where it is running and unique in that across the entire system, the service's name corresponds only to that service. This does not mean different logical services in different locations; a typical usage of Domain Name System (DNS) where each service is assigned its own DNS name would satisfy this recommendation. Consistent names (identities) for services are required so that the system policy is manageable.	*TSB relies on the underlying compute platform to generate identities, but applying consistent naming across every cluster as a best practice conforms to this recommendation.*	N/A

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SM-DR12: SIGNING CERTIFICATE DEPLOYMENT The Service Mesh control plane's certificate management system should have its ability to generate self-signed certificates disabled. This functionality is frequently used to bootstrap an initial signing certificate for all other identity certificates in the Service Mesh. Instead, the signing certificate used by the mesh's control plane should always be rooted in the enterprise's existing PKI's root of trust and provided securely to the Service Mesh control plane at startup. This simplifies the management of those certificates by an existing PKI (e.g., for revocation or audit). Further, we recommend that separate intermediate signing certificates should be generated for different domains to simplify rotation and enable fine-grained revocation.	*We recommend issuing an intermediate signing certificate for Istio from your existing PKI. Use that intermediate certificate to issue a signing certificate to each Istio control plane instance in that environment, and have Istio operate as usual to issue workload (leaf) certificates to all pods in each cluster.*	N/A
SM-DR13: IDENTITY CERTIFICATE ROTATION The lifetime of a microservice's identity certificate should be as short as is manageable within the infrastructure—preferably on the order of hours. This helps limit attacks since an attacker can only use a credential to impersonate a service until that credential expires, and successfully re-stealing a credential increases the difficulty for an attacker.	*Istio securely provisions strong identities to every workload with X.509 certificates and automates key and certificate rotation at scale to limit attacks in time. Certificate lifetimes may be configured to last hours or less and are automatically updated.*	TSB Istio

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SM-DR14: CYCLE CONNECTIONS ON IDENTITY CHANGE When a service proxy's identity certificate is rotated, the service proxy should efficiently retire existing connections and establish all new connections with the new certificate. Certificates are only validated during the mTLS handshake, so replacing existing connections when a new certificate is issued is not strictly required; rather, this is important for limiting attacks in time.	*As mentioned in SM-DR5, connections are cycled regularly to ensure connection policy is applied regularly, including mutual authentication during connection handshake. Similarly, as required by SPIFFE, Istio cycles connections when workload identity changes. Envoy supports connection draining to facilitate graceful certificate updates*	TSB Istio
SM-DR15: NON-SIGNING IDENTITY CERTIFICATES Certificates used to identify microservices should not be signing certificates.	*The service identity certificates issued by Istio are not signing certificates. The Istio control plane issues SPIFFE IDs for all workloads. SPIFFE specifies that all identity certificates be leaf certificates, not signing certificates.*	TSB Istio

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SM-DR16: WORKLOAD AUTHENTICATION BEFORE CERTIFICATE ISSUANCE The Service Mesh control plane's certificate management system should perform authentication of a service instance before issuing it an identity certificate. In many systems, this can be achieved by attesting the instance against the system's orchestration engine, and by using other local proofs (e.g. secrets retrieved from an HSM). The same care should be taken in provisioning the signing certificate for the Service Mesh control plane's certificate management system. That signing certificate should be retrievable only by the Service Mesh control plane and only after some form of attestation has been done against it.	*In Kubernetes, Istio's Envoy sidecar proxies use the local service account token to authenticate service instances with the control plane. On platforms without a service identity, Istio can use other identities that can group workload instances, such as service names. For workloads on such platforms, e.g., in EC2, Azure Virtual Machines, etc., Tetrate Service Bridge automatically translates specific platform credentials into service identities so they can join the mesh.*	TSB Istio
SM-DR17: SECURE NAMING SERVICE If the certificate used for mTLS carries server identity, then the Service Mesh should provide a secure naming service that maps the server identity to the microservice name that is provided by the secure discovery service or DNS. This requirement is needed to ensure that the server is the authorized location for the microservices and to protect against network hijacking.	*Istio's secure naming facility maintains a mapping between server identities and server names, built from the Kubernetes service entry configuration. The control plane watches the Kubernetes API server, generates the secure naming mappings, and distributes them securely to the PEPs. This allows clients to verify that a particular server is authorized to run a given service.*	TSB Istio Kubernetes

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SM-DR18: GRANULAR IDENTITY Each microservice should have its own identity, and all instances of this service should present the same identity at runtime. This allows for access policy at the level of microservice in a given namespace. This is required since common microservice runtimes default to issuing identities per namespace rather than per service so that all services in the same namespace present the same runtime identity unless otherwise specified. In addition, labels can be used to augment the service name (identity) to enable granular logging configuration and to support granular authorization policies.	*Rather than use the default service account, Istio provides unique identities for each service (by way of SPIFFE identity certificates). This allows authorization policies to be authored at the service level, in addition to the coarser-grained global and namespace levels. Individual instances of a particular service share the common service identity.*	TSB Istio
SM-DR19: AUTHENTICATION POLICY SCOPE The feature to specify the policy scope for authentication should have the following minimal options: (a) all microservices in all namespaces, (b) all microservices in a particular namespace, and (c) a specific microservice in a given namespace (using the runtime identity referred to in SM-DR17).	*TSB enables flexible authentication policy authorship that provides a superset of the recommended scope objects, including TSB-specific abstractions such as tenants, workspaces, groups, and applications.*	TSB Istio

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SM-DR20: AUTHENTICATION TOKEN Tokens should be digitally signed and encrypted so that claims included in them have the assurance of authenticity since these claims can be used to augment or be part of an authenticated identity to build access control decisions. Further, these tokens must only be passed by loopback device (to ensure that there is no network path involved) or through an encrypted channel.	*TSB uses JSON web tokens (JWTs) for end-user authentication and for authentication between TSB components to satisfy the recommendation.*	TSB Istio

4.5: Configuration for Monitoring Capabilities

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SM-DR21: LOGGING EVENTS The proxy should log input validation errors and extra (unexpected) parameters errors, crashes, and core dumps. Common attack detection capabilities should include bearer token reuse attack and injection attacks.	*Envoy produces access logs to record the recommended parameters and can be configured to collect core dumps if needed. Tetrate Service Bridge can feed data from all applications into attack detection systems, but does not implement attack detection itself.*	TSB Istio
SM-DR22: LOGGING REQUESTS The proxy should log at least the Common Log Format fields for irregular requests (e.g., non-200 responses when using HTTP). Logging for successful requests (e.g., 200 responses) tends to be of little value when metrics are available.	*By default, Istio collects Envoy's default access logging, and may be configured to be more or less verbose. Tetrate Service Bridge facilitates collecting and correlating logging and telemetry data for all applications, across clusters, clouds, and data centers.*	TSB Istio
SM-DR22.1: LOG MESSAGE CONTENT Log messages should contain, at a minimum, the event date/time, microservice name or identity, request trace id, message, and other contextual information (e.g., requesting user identity and URL). However, logging should take care to mask sensitive information, for example Bearer tokens.	*In addition to Envoy's access logs, Istio also collects metrics and distributed tracing data, observable through TSB. Istio does not collect sensitive information in the clear.*	TSB Istio

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SM-DR23: MANDATORY METRICS The configuration for gathering metrics using Service Mesh for external client and microservice calls should involve, at a minimum, the following: (a) the number of client/service requests in a given duration, (b) the number of failed client/service requests by failure code, and (c) the average latency per service as well as the average total latency per complete request lifecycle (ideally as a histogram; also by failure code).	*Istio generates a set of service metrics based on the four "golden signals" of monitoring: latency, traffic, errors, and saturation. Istio also provides detailed metrics for the mesh control plane. Tetrate Service Bridge facilitates collecting and correlating logging and telemetry data for all applications, across clusters, clouds, and data centers.* *Istio's standard metrics include: HTTP request count, duration (distribution), and size (distribution); response size (distribution); gRPC request and response message count. For TCP traffic specifically: bytes sent and received (counter) and connections opened and closed (counter).*	TSB Istio
SM-DR24: IMPLEMENTING DISTRIBUTED TRACING When configuring the service proxies for implementing distributed tracing, care should be taken to ensure that the application services are instrumented to forward the headers for communication packets they received.	*Istio initiates and facilitates distributed tracing, augmenting traces with Envoy. Note, however, that applications must participate by propagating the trace context between incoming and outgoing requests.*	TSB Istio

4.6: Configuration for Network Resilience Techniques

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SM-DR25: STORING DATA FOR IMPLEMENTATION OF NETWORK RESILIENCE Data pertaining to retries, timeouts, circuit breaking settings, or canary deployments (in general, all control plane configuration data) should be stored in robust data stores, such as Key/Value stores.	*Data configuring resiliency settings is stored securely as custom resource definitions (CRDs) in Kubernetes.*	TSB Istio
SM-DR26: IMPLEMENTATION OF HEALTH CHECKING OF SERVICE INSTANCES The health checking function for service instances should be tightly integrated with the service discovery function to maintain the integrity of the information used for load balancing.	*Istio tightly integrates with Kubernetes built-in health checking as well as its own passive health checks to maintain the integrity of its service registry. (See also MS-SS-7.)*	TSB Istio Kubernetes

4.7: Configuration for Cross-Origin Resource Sharing (CORS)

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SM-DR27: CORS An edge service (i.e., entry point for microservice) may often have to be configured for CORS for communicating with external services, such as a web UI client service. The CORS policy for an edge service should be configured using the Service Mesh capability (e.g., VirtualService resource's CorsPolicy configuration in Istio) rather than handling it through the microservice application service code.	*TSB exposes configuration for Envoy to send CORS headers which, by default, are maximally strict. More permissive CORS policy may be explicitly configured.*	TSB Istio

4.8: Configuration of Permissions for Administrative Operations

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SM-DR28: ACCESS CONTROL FOR ADMINISTRATIVE OPERATIONS There should be granular access control permissions for all administrative operations in a service mesh (e.g., policy specifications, configuration parameters for service resiliency parameters, canary deployments, retries, etc.) that are specifiable at the level of all services in a namespace, a named service within a namespace, etc. Generally, the interface for exercising this functionality may not be part of the service mesh itself but part of an installation software or orchestration software used for configuring the application service cluster.	*Fine-grained, hierarchical permissions are a core value proposition of Tetrate Service Bridge. TSB maintains a resource hierarchy, including tenants and workspaces, which—when combined with users, groups, and roles synchronized with your organization's directory services—can be used to author granular access control policy and auditability for all administrative operations. TSB offers the ability to propagate and merge security settings down its resource hierarchy such that security teams can specify core security policy at the most appropriate level in the hierarchy, and TSB will propagate and merge the configuration with tenant and workspace security settings further down the tree.*	TSB

SECTION 2

NIST SP 800-204B

ATTRIBUTE-BASED ACCESS CONTROL FOR MICROSERVICES-BASED APPLICATIONS USING A SERVICE MESH

SB 800-204B, the third paper in the series, offers a deep dive into best practices for implementing fine-grained and dynamic authentication and authorization for microservices using attribute-based access control (ABAC).

4.2: Service Mesh Configuration

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
ISMC-SR-1 If certificate-based authentication is used for authenticating service calls, the signing certificate used by the service mesh's CA module should be rooted in the organization's existing Public Key Infrastructure (PKI) to allow for auditability, rotation, and revocation.	*We recommend issuing an intermediate signing certificate for Istio from your existing PKI. Use that intermediate certificate to issue a signing certificate to each Istio control plane instance in that environment, and have Istio operate as usual to issue workload (leaf) certificates to all pods in each cluster. (See also SM-DR12 above.)*	N/A
ISMC-SR-2 Communication between the service mesh control plane and the hosting platform's configuration server must be authenticated and authorized.	*Istio validates a node-local proof of identity—based on service account or similar—at the workload for Envoy proxies to hand to the Istio control plane. The control plane serves TLS to connecting Envoy proxies which, in turn, offer that proof of identity. Once authenticated and authorized, the control plane then sends configuration to the given proxy instance, including an identity that may be used to establish mTLS connections with other parts of the system.*	TSB Istio

4.3: Higher-Level Security Configuration Parameters

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
AHLC-SR-1 Containers and applications should not be run as root (thus becoming privileged containers).	*The configuration and deployment of application containers and other application workloads is managed by the underlying compute platform and, as such, is outside the scope of TSB. TSB components themselves do not run as privileged containers.*	N/A
AHLC-SR-2 Host path volumes should not be used, because they create tight coupling between the container and the node on which it is hosted, constraining the migration and flexible resource scheduling process.	*See above.*	N/A
AHLC-SR-3 Configure the container file system as read-only by default for all applications, overriding only when the underlying application (e.g., database) must write to disk.	*See above.*	N/A

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
AHLC-SR-4 Explicitly prevent privilege escalation for containers.	*See above.*	N/A

4.4.2: Service-Level Authentication

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SAUN-SR-1 A policy object relating to service-level authentication should be defined that requires mTLS be used for communication. The policy object should be expressive enough to be defined at various levels (given below) with features for overrides at the lower levels or inheritance of the requirement specified at the higher levels. The following are the minimum required levels: • Global level or the service mesh level • Namespace level • Workload or microservices level, used for applying authentication and authorization policies for a subset of traffic to a subset of resources (e.g., particular microservices, hosts or ports) • Port level, taking into account that certain traffic is designed for communicating through designated ports.	*Istio offers the ability to configure policy globally, at the namespace level, and at individual services. Tetrate Service Bridge allows such policies to be defined and enforced consistently across clusters, making sure that as clusters come and go dynamically, policy is applied at the right level.* *TSB's resource hierarchy can be used to author granular policy, including mTLS requirements, at multiple levels. TSB offers the ability to propagate and merge policy settings down its resource hierarchy so that security teams can specify core security policy (like mTLS requirements) at the most appropriate level in the hierarchy. TSB will propagate and merge configuration from higher in the tree with security settings further down the tree. In addition to global, namespace, workload, and port levels, TSB's resource hierarchy also offers policy configuration at application, cluster, workspace, and tenant levels. (See also SM-DR28 above.)*	TSB Istio

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SAUN-SR-2 If the certificate used for mTLS carries server identity, then the service mesh should provide a secure naming service that maps the server identity to the microservice name that is provided by the secure discovery service or DNS. This requirement is needed to ensure that the server is the authorized location for the microservices and to protect against network hijacking.	*Istio's secure naming facility maintains a mapping between server identities and server names, built from the Kubernetes service entry configuration. The control plane watches the Kubernetes API server, generates the secure naming mappings, and distributes them securely to the PEPs. This allows clients to verify that a particular server is authorized to run a given service. (See also SM-DR17 above.) Tetrate Service Bridge also acts as a naming provider.*	TSB Istio

4.4.3: End User Authentication

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
EAUN-SR-1 A request authentication policy must, at the minimum, provide the following information and must be enforced by the sidecar proxy: • Instructions for extracting the credential from the request • Instructions for validating the credential	*As for SAUN-SR-1 above, Istio offers these capabilities out of the box and Tetrate Service Bridge ensures consistency across clusters. TSB's resource hierarchy can also be used to author granular policy at multiple logical levels that cascade down the hierarchy tree.*	TSB Istio

4.5.1: Service-Level Authorization Policies

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
SAUZ-SR-1 A policy object describing service-to-service access should be in place for all services in the mesh. At a minimum, these policies should restrict access to the namespace level (e.g., “services in namespace A can call services in namespace B”). Ideally policies should restrict access to individual services (e.g., "service Foo in namespace A can call service Bar in namespace B").	*Istio offers a workload selector field to determine the scope of a particular policy. The selector may specify to apply a policy object mesh-wide, namespace-wide, specific to a particular workload, or to a set of workloads with matching labels. Tetrate Service Bridge enhances policy expressiveness with resource hierarchies as described in SAUN-SR-1 and EAUN-SR-1 above, allowing for more articulate policy with less configuration. For example, TSB enables authorship of a global policy such that all workloads are authorized to connect only to other workloads in their namespace, which may then be further restricted at the workspace or application level—which is not possible to express with a single policy object in Istio alone.*	TSB Istio

4.5.2: End-User Level Authorization Policies

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
EUAZ-SR-1 When a sidecar communicates with an authentication or authorization system, that communication must be secured with either the mesh's built-in service-to-service authentication and authorization capabilities or using an existing enterprise Identity and Access Management (IAM) that is not part of the service mesh.	*While the disposition of an external authentication and authorization system is out of the scope of Tetrate Service Bridge, TSB offers the capability to configure security settings for communications from services in the mesh to external systems. We also recommend deploying IAM systems in the mesh to take advantage of the built-in security benefits.*	N/A
EUAZ-SR-2 The sidecar should generate logs for every service request to ensure that authentication and authorization policies are enforced and relay telemetry data for the generation of metrics to ensure no degradation of service that will impact availability.	*Envoy produces access logs and telemetry data to record the required parameters, including whether the request was denied due to policy or similar reasons. Additional logging and metrics regarding enforcement disposition is configurable. Tetrate Service Bridge offers central configuration for logging and metrics. It also relays logs, telemetry, and tracing data from all applications to a central store for use in the TSB management plane dashboards, audit, and to feed external analysis systems.* *See also NIST SP 800-53 AU-3, AU-9, and AU-12.*	TSB Istio

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
EUAZ-SR-3 All application traffic should carry end user credentials, and there should be a policy in the mesh enforcing that credentials are present.	*Istio offers the configuration and enforcement of policy to validate end-user credentials in a single cluster. As for SAUN-SR-1, EAUN-SR-1, and SAUZ-SR-1 above, Tetrate Service Bridge enhances policy expressiveness with resource hierarchies allowing for more articulate policy with less configuration. Tetrate Service Bridge also provides a centralized management plane where such policies may be defined and consistently enforced across clusters, clouds, and on-premises.*	TSB Istio

4.6: Authorization Policy Elements

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
APE-SR-1 The authorization policy should, at the minimum, contain the following policy elements: • Policy types – Positive (ALLOW) or Negative (DENY) • Policy target or authorization scope – the namespace, a particular service (application name), and version • Policy sources – covers the set of authorized services • Policy operations – specifies the operations on the target resources that are covered under the policy • Policy conditions – the metadata associated with the request that must be met for the application or invocation of the policy	*Istio offers authorization policy configuration to satisfy this recommendation. As above, Tetrate Service Bridge makes it easier to manage policy consistently and dynamically across your entire infrastructure.*	TSB Istio

NIST RECOMMENDATION	TETRATE IMPLEMENTATION NOTES	SUPPORT
APE-SR-2 The policy should cover all of the operations that are part of the application type. For example, if the application is implemented as a REST API, all of the operations (also called HTTP verbs or HTTP methods) that are part of the REST API must be included: POST: This is equivalent to creating a resource. GET: This is equivalent to reading the contents of the resource. PUT: This is equivalent to updating the resource by replacing. PATCH: This is equivalent to updating the resource by modifying. DELETE: This is equivalent to deleting the resource.	*Per the recommendation, Istio provides for policy articulation and enforcement at the application layer (Layer 7). As above, Tetrate Service Bridge facilitates managing policy consistently and dynamically across your entire infrastructure.*	TSB Istio
APE-SR-3 A default policy should be authored in the system that rejects all requests that are unauthenticated, mandates that service and end-user credentials be present on every request, restricts all communication to services within the application's own namespace, and allows service communication across namespaces only through an explicit policy.	*Istio supports authoring and enforcing the recommended default policy. As above, Tetrate Service Bridge makes it easier to manage that policy consistently and dynamically across your entire infrastructure.*	TSB Istio

tetrate

About Tetrate

Started by Istio founders to reimagine application networking, Tetrate is the enterprise service mesh company managing the complexity of modern, hybrid cloud application infrastructure. Its flagship product, Tetrate Service Bridge, provides an edge-to-workload application connectivity platform to deliver business continuity, agility, and security for enterprises on the journey from traditional monoliths to the cloud. Customers get consistent, baked-in observability, runtime security and traffic management in any environment. Tetrate remains a top contributor to the open source projects Istio and Envoy Proxy.

691 S Milpitas Blvd, Suite 217, Milpitas, CA 95035, USA

www.tetrate.io | info@tetrate.io