NIST 800-53 Compliance for Containers and Cloud

Contents

Introduction

The National Institute of Standards and Technology (NIST) created the Cybersecurity Framework in 2014 as a standardized set of security and operational requirements for private sector organizations that conduct business over the internet with, or on behalf of, the federal government. While the government rarely gets credit for innovation, the Framework has come to define a new way of encouraging more efficient and secure government services. It also encourages more interaction among the government and private industries, good for technology and for business.

NIST has intended the Framework and related policies to be a roadmap for security, but it has enabled something far beyond risk mitigation. It's notable to see how the Framework has helped U.S businesses reach new markets and become more competitive globally as a result of their adherence to NIST compliance standards.

NIST 800-53 and NIST 800-171 compliance standards combine guidelines for leveraging government relationships and using sensitive data, but also ensure the security of all that work and data. These standards address manufacturers and developers of hardware, software, and systems who touch sensitive data anywhere along the data supply chain. The result is basically a checklist, but the inherent wisdom is that organizations are able to check against these to ensure their compliance status, and do so in a game with a continuous scoring system; getting out of compliance can happen because of unintended changes to database configurations or permissions to cloud resources, etc. Without standards like NIST 800-53 and 800-171, businesses could be inadvertently putting critical data at risk.

Where does NIST 800-53 apply?

NIST 800-53 represents a comprehensive set of controls broken down according to families, major controls, and sub controls. Many of these controls are interdependent and rely on adherence with one to impact others. With 800-53, organizations are able to work according to FedRAMP regulations, which keeps them operational at both a technology and business level.

NIST 800-171 was designed to help non-federal entities, like contractors, implement security systems and practices in order to protect Controlled Unclassified Information (CUI) in non-Federal systems. The framework sets forth a standardized set of requirements for CUI security, so contractors can do their part to protect and safeguard confidential information.

Adhering to the Framework, and these standards in particular, means that an organization has built their own security around a single reference that uses the comprehensive knowledge of hundreds of governmental agencies, all of which are required to use demanding security best practices. While it requires considerable effort to be compliant with the Framework, once achieved, the organization can tout compliance with the myriad standards, governance policies, audit checklists, and other aspects of critical security

necessary for working with almost any organization that mandates strict security adherence. Organizations that use the Framework benefit by being prepared for almost any security requirements demanded by their industries, the government, or their own customers.

These standards also act as a safety net that will ultimately keep a business in operation (because being out of compliance means losing the ability to use this sensitive data), but will also maintain the integrity of sensitive data.

The Framework is often cited as a model of flexibility, certainly key to both its adoption, and strength as a security model. It takes into account that cybersecurity is not a one-size-fits-all proposition and guides users through examples that demonstrate the reality of operating in cloud environments where change happens constantly. These standards recognize that the technology necessary to be competitive in today's global market requires constant change, dynamic integration with different applications and resources, and options for crowdsourcing changes.

Three are five basic principles that govern the NIST framework: Identify, Protect, Detect, Respond, and Recover. These NIST cybersecurity standards support that and give users the necessary guidelines to keep sensitive data secure and flexible.

What's new in Revision 5?

SP 800-53 Revision 5 was published on Sept. 23, 2020, after being a draft document since 2016. It supersedes Revision 4, which was published in 2015 and will be officially withdrawn on Sept. 23, 2021. Organizations looking to be compliant with NIST 800-53 will typically consider adopting the new standard six months after the publication of the new revision.

Revision 5 updates control descriptions, so instead of centering on the responsible agent, the focus is on the expected outcome. All individual controls are given a distinctive name, and information security and privacy are consolidated as a single goal. Two new families are introduced: *Processing and Transparency* and *Supply Chain Risk Management*. Baseline controls are moved to a new document, NIST SP 800-53B, specific for federal agencies, so other organizations can implement their own baselines.

According to NIST, this update delivers tighter relationships among disparate systems, components, applications, hardware, and digital services with an emphasis on building resilience into the economic, military, and human services that are critical to defending the national security interests of the country.

The most notable updates in Revision 5 include:

- **Control catalog consolidation:** Privacy and security controls are integrated into a single control catalog.
- **Supply chain risk management integration:** A new control family is now being used for supply chain risk management (SCRM) and is integrated throughout the catalog.
- **Addition of practice controls:** Based on the latest threat intelligence and cyberattack information, these controls support cybersecurity readiness, secure system design, governance, and accountability.

- **Outcome-based controls approach:** Rev. 5 does this by removing the entity responsible for satisfying the control from the control statement.
- **Content relationship descriptions:** Provides clarification of the relationship between requirements and controls, and the interaction of security and privacy controls.
- **Separates the control selection process from controls:** This enables controls to be used by different communities of interest, including engineers, SecOps, DevOps, architects, security engineers, and business line owners.
- **Transfers control baselines and tailors guidance to NIST SP 800-53B:** This content has moved to the new (draft) *Control Baselines for Information Systems and Organizations*.

How to comply with NIST 800-53 when running containers and Kubernetes

NIST 800-53 is defined as, "...a catalog of security and privacy controls to protect organizational operations and assets, individuals, other organizations, and the Nation from a diverse set of threats and risks, including hostile attacks, human errors, natural disasters, structural failures, foreign intelligence entities, and privacy risks." As such, it covers controls for a wide variety of resources, environments, and assets. While the controls themselves are not unique to specific technologies, there is a "translate" capability within the framework that guides security and compliance teams in mapping controls to containers or Kubernetes security.

With containers, developers are largely responsible for the security of apps and images instead of the operations team. This change in responsibilities often requires much greater coordination and cooperation among personnel than was previously necessary.

These are the control families within NIST 800-53 that are most relevant to containers and Kubernetes:

Access controls

Control Summary: How user accounts are generated, secured, controlled, and authorized.

For containers and Kubernetes: Implement detections for processes or users trying to break beyond the security constraints of their assigned user and service accounts. Do it at container image or runtime level.

Audit and accountability

Control Summary: How audits of activity are generated and stored, and how alerts are triggered, reviewed, and responded to.

For containers and Kubernetes: Set up tools that enable auditing Kubernetes events and general security issues. Generate reports that show the general security state of infrastructure.

System and communications protection

Control Summary: How to implement cypher mechanisms and prevent counterfeit data and components.

For containers and Kubernetes: Implement mechanisms to detect and counter attempts to impersonate or modify legitimate systems inside containers, detecting drifts from container images on runtime.

Configuration management

Control Summary: How the baseline configuration of a system is kept under configuration control, and how changes to it are tested before implementation.

For containers and Kubernetes: Test container images for security and misconfiguration issues. Do it in CI pipelines, or right before they are deployed.

System and information integrity

Control Summary: Mechanisms for identification, correction, and reporting of system flaws.

For containers and Kubernetes: Implement a centralized pipeline for CI/CD with image scanning. Install an admission controller that can intercept deployments that bypass the pipeline. Set up runtime detections for abnormal behavior that can only be detected while a container is running. Implement notification and reporting mechanisms.

System and services acquisition

Control Summary: Quality metrics and guarantees from developers about the systems they provide.

For containers and Kubernetes: Similar to the previous point, focus on checking for misconfigurations and vulnerabilities on the software deployed in containers.

Incident response

Control Summary: Incident response policy characteristics and how incidents are handled.

For containers and Kubernetes: Ensure security events triggered in containers and Kubernetes are correctly notified and filtered, and that the information provided facilitates incident response tasks.

Security assessment and authorization

Control Summary: How security compliance checks are done before internal systems connect with each other.

For containers and Kubernetes: Map security checks against specific controls and families of NIST 800-53 inside your security tool.

Any enterprise looking to achieve continuous NIST 800-53 compliance should start by incorporating automation of compliance and container configuration standards into their security approach to meet their security and compliance requirements in adherence with NIST SP 800-53.

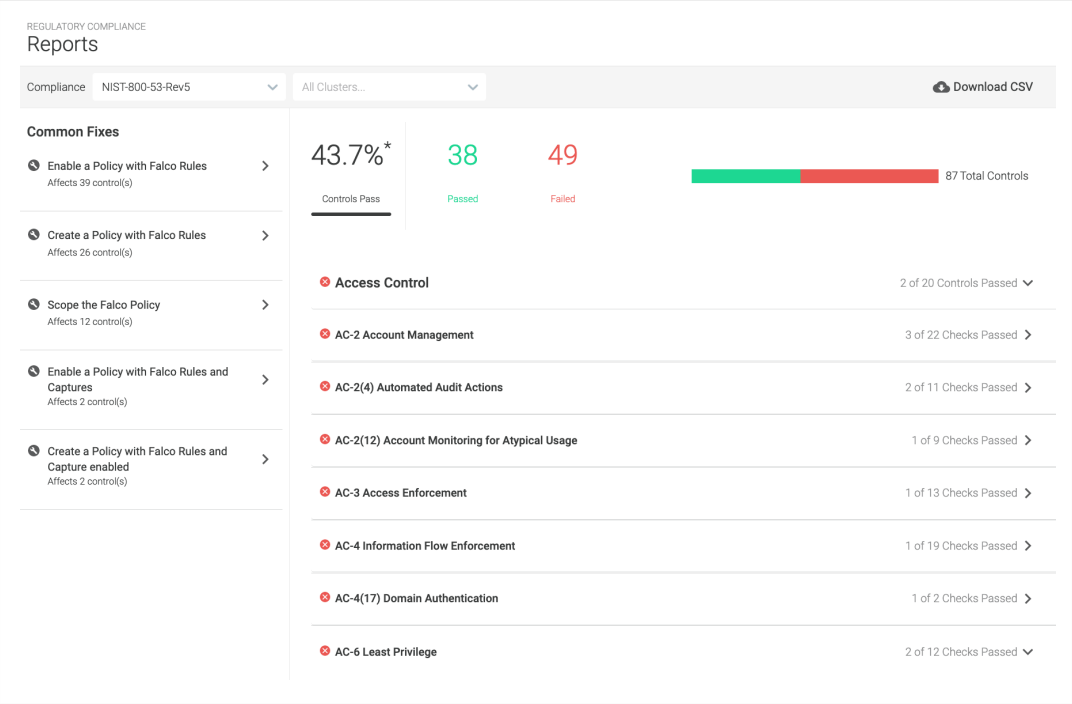
How can Sysdig help?

With Sysdig, your security and DevOps teams can validate NIST 800-53 compliance across containers and cloud. Sysdig provides out-of-the-box compliance checks for container and Kubernetes environments that provide an instant snapshot of compliance posture, highlighting each passed and failed compliance control – with drill-down capabilities to identify the clusters, namespaces, and deployments failing the checks. Teams can also customize policies based on specific compliance and governance requirements. Sysdig performs a continuous audit of container infrastructure events to facilitate NIST 800-53 compliance and corresponding incident response. Sysdig also captures detailed container forensics data that serve as a proof of compliance for third-party auditors even after containers are gone.

NIST 800-53 Controls Supported by Sysdig

Control Group AC: Access Control

This group provides a framework for access control policies and procedures.



Control AC-2: Account Management

Assigns account managers for specific systems; defines approval processes for access and authorizations; monitors account activity; notifies when accounts are inactive, or no longer required; notifies when users are terminated or transferred; provides authorization based on valid access authorization, usage, and business function.

Control ID	Control Name	How Sysdig helps
AC-2	Account Management	Identify binaries used to manage users, passwords, or permissions. Exclusions include: Unix sudo and su, any activity within containers, command lines that do not change status of users, passwords, or permissions. Detect new SSH host connections not already connected to an allowed group of hosts. Specific applications monitored for attempts to spawn shells below non-shell applications. Detect any attempts to run an interactive command by a system (non logged-in) user. Detect a console login without using multi-factor authentication (MFA). Detect a user login without using multi-factor authentication (MFA). Detect when multi-factor authentication (MFA) configuration for all access has been deactivated. Detect Kubernetes operations by non-allowed users. Detect requests made by allowed, anonymous users. Detect any attempt to create a serviceaccount in the kube-system or kube-public namespaces. Detect any attempt to modify/delete a system ClusterRole/Role starting. Detect any attempt to create a ClusterRoleBinding to the cluster-admin user. Detect any attempt to create a Role/ClusterRole with wildcard resources or verbs. Detect any attempt to create a Role/ClusterRole that can execute to pods. Detect attempts to: • Create Kubernetes service accounts. • Delete Kubernetes service accounts. • Create Kubernetes clusters and/or roles. • Delete Kubernetes clusters and/or roles. • Create a clusterrolebinding. • Delete a clusterrolebinding.

		Detect Kubernetes operations by a user name that may be an admin with full access. Identify binaries used to manage users, passwords, or permissions. Exclusions include: Unix sudo and su, any activity within containers, command lines that do not change status of users, passwords, or permissions. Communicates that a user must be specified to run instead of default (root).
AC-2 (4)	Automated Audit Actions	Enable automated run-time policies using the Falco rules defined specifically for account activity. Detect users added to AWS groups. Detect the creation of a new AWS user. Identify when and where a terminal shell enters and exits a container. Identify binaries used to manage users, passwords, or permissions. Exclusions include: Unix sudo and su, any activity within containers, command lines that do not change status of users, passwords, or permissions. Detect attempts to attach to, or execute, a pod. Detect attempts to: • Create Kubernetes service accounts. • Delete Kubernetes service accounts. • Create Kubernetes clusters and/or roles. • Delete Kubernetes clusters and/or roles. • Create a clusterrolebinding. • Delete a clusterrolebinding. Identify when and where an attached terminal shell enters and exits a container. Identify binaries used to manage users, passwords, or permissions. Exclusions include: Unix sudo and su, any activity within containers, command lines that do not change status of users, passwords, or permissions.
AC-2 (12)	Account Monitoring / Atypical Usage	Detect AWS execution commands in unused regions. Detect when an inline policy that allows access to all resources has been added to a group. Detect when an IAM policy that allows all has been created.

		Detect Kubernetes operations by non-allowed users. Detect requests made by allowed, anonymous users. Detect any attempt to create a serviceaccount in the kube-system or kube-public namespaces. Detect any attempt to modify/delete a system ClusterRole/Role starting. Detect any attempt to create a ClusterRoleBinding to the cluster-admin user. Detect any attempt to create a Role/ClusterRole with wildcard resources or verbs. Detect any attempt to create a Role/ClusterRole that can execute to pods. Detect Kubernetes operations by a user name that may be an admin with full access. Communicates that a user must be specified to run instead of default (root).

Control AC-3: Access Enforcement

Establishes the enforcement policies and procedures for approved authorizations for logical access to information and system resources in accordance with applicable access control policies.

Control ID	Control Name	How Sysdig helps
AC-3	Access Enforcement	Identify binaries used to manage users, passwords, or permissions. Exclusions include: Unix sudo and su, any activity within containers, command lines that do not change status of users, passwords, or permissions. Detect attempts to: • Create Kubernetes service accounts. • Delete Kubernetes service accounts. • Create Kubernetes clusters and/or roles. • Delete Kubernetes clusters and/or roles. • Create a clusterrolebinding. • Delete a clusterrolebinding.

		Detect any attempt to create a serviceaccount in the kube-system or kube-public namespaces. Detect any attempt to modify/delete a system ClusterRole/Role starting. Detect any attempt to create a ClusterRoleBinding to the cluster-admin user. Detect any attempt to create a Role/ClusterRole with wildcard resources or verbs. Detect Kubernetes operations by a user name that may be an admin with full access. Enforces approved authorizations for logical access to information and system resources in accordance with applicable access control policies.

Control AC-4: Information Flow Enforcement

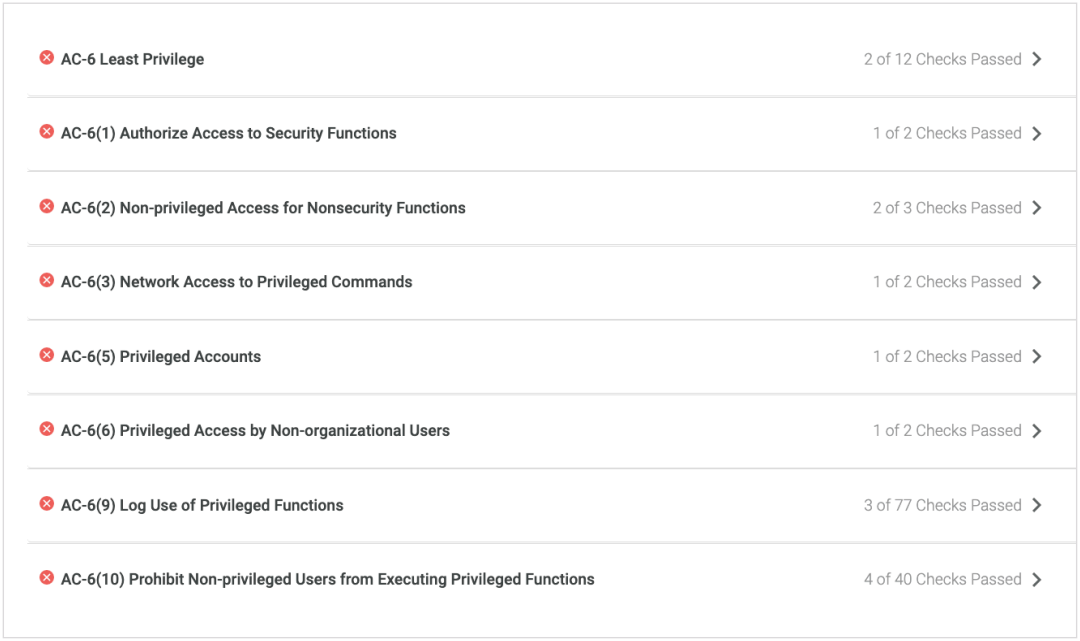
Enforces approved authorizations for controlling flow of information within the system and between connected systems.

Control ID	Control Name	How Sysdig helps
AC-4	Information Flow Authorizations	Detect any outbound connection to a destination outside of an allowed set of IPs, networks, or domain names. Detect any inbound connection from a source outside of an allowed set of IPs, networks, or domain names. Detect an attempt to run a program with a disallowed HTTP_PROXY environment variable. Detect unexpected traffic, specifically, UDP traffic not on port 53 (DNS) or other commonly used ports. Detect attempts to contact the EC2 instance metadata service from a container. Detect attempts to contact the Cloud Instance Metadata Service from a container. Detect attempts to contact a Kubernetes API server from a container. Detect attempts to use Kubernetes NodePorts from a container.

		Detect traffic to images outside the local subnet. Detect traffic that is not to authorized specific server processes and ports. Detect if a public IP address has been allocated to an account. Detect if a public IP address has been associated with a network interface. Detect creation of an AWS internet-facing load balancer. Detect an attempt to start a pod using the host network. Detect an attempt to start a service with a NodePort service type.
AC-4 (17)	Domain Authentication	Detect when an ingress object without a TLS certificate has been created.

Control AC-6: Least Privilege

Establishes the application of the principle of least privilege for specific systems and actions.



Control ID	Control Name	How Sysdig helps
AC-6	Employ Least Privilege	Specific applications monitored for attempts to spawn shells below non-shell applications. Detect requests made by allowed, anonymous users. Detect any attempt to create a serviceaccount in the kube-system or kube-public namespaces. Detect any attempt to modify/delete a system ClusterRole/Role starting. Detect any attempt to create a Role/ClusterRole with wildcard resources or verbs. Detect any attempt to create a Role/ClusterRole that can execute to pods. Detect any attempt to run an interactive command by a system (non logged-in) user. Detect when an administrator policy has been attached to a user. Detect an attempt to start a pod with a privileged container. Detect an attempt to start a pod with a volume from a sensitive host directory.
AC-6 (1)	Authorize Access to Security Functions	Detect requests made by allowed, anonymous users.
AC-6 (2)	Non-Privileged Access for Non-Security Functions	Detect Kubernetes operations by a user name that may be an admin with full access. Communicates that a user must be specified to run instead of default (root).
AC-6 (3)	Network Access to Privileged Commands	Detect requests made by allowed, anonymous users.
AC-6 (5)	Privileged Accounts	Detect requests made by allowed, anonymous users.
AC-6 (6)	Privileged Access by Non-Organizational Users	Detect requests made by allowed, anonymous users.
AC-6 (8)	Privilege Levels for Code Execution	Identify binaries used to manage users, passwords, or permissions. Exclusions include: Unix sudo and su, any activity within containers, command lines that do not change status of users, passwords, or permissions. Detect attempts to modify shell configuration files.

		Detect attempts to read shell configuration files by non-shell programs. Detect a root user executing an AWS command.
AC-6 (9)	Log Use of Privileged Functions	Identify binaries used to manage users, passwords, or permissions. Exclusions include: Unix sudo and su, any activity within containers, command lines that do not change status of users, passwords, or permissions. Detect any attempt to create a Role/ClusterRole that can perform write-related actions. Identify binaries used to manage users, passwords, or permissions. Exclusions include: Unix sudo and su, any activity within containers, command lines that do not change status of users, passwords, or permissions. Detect an attempt to write to any file directly below "/" or "/root." Identify when a database-server related program spawns a new process other than itself. Detect an attempt to change a program or thread's namespace (commonly done as a part of creating a container) by calling setNS. Detect the initial process started in a privileged container. Exceptions are made for known trusted images. Detect the initial process started by a container that has a mount from a sensitive host directory. Exceptions are made for known trusted images. Detect the initial process started by a container that is not in a list of allowed containers. Detect any network activity performed by system binaries that are not expected to send or receive network traffic. Detect an attempt to change users by calling setuid. Excluded: a) sudo/su and b) users "root" and "nobody" suing to itself. Identifies when any files below "/dev" are created, with the exception of known programs that manage devices. Detect when a Netcat Program runs inside a container that allows remote code execution.

		Detect when network tools are launched inside of a container. Detect when network tools are launched on a host. Detect setuid or setgid bits set via chmod. Detect symlink that has been created over sensitive files. Detect new packet socket at the device driver (OSI Layer 2) level in a container. Detect when stdout/stdin are redirected to a network connection in a container. Detect when a new executable is created in a container due to chmod. Detect when a new executable is created in a container due to open+create. Detect a root user executing an AWS command. Detect an attempt to start a pod with a privileged container. Detect an attempt to start a pod with a volume from a sensitive host directory. Detect an attempt to create a pod in the kube-system or kube-public namespaces.
AC-6 (10)	Prohibit Non-Privileged Users From Executing Privileged Functions	Identify binaries used to manage users, passwords, or permissions. Exclusions include: Unix sudo and su, any activity within containers, command lines that do not change status of users, passwords, or permissions. Detect any attempt to create a Role/ClusterRole that can perform write-related actions. Identify binaries used to manage users, passwords, or permissions. Exclusions include: Unix sudo and su, any activity within containers, command lines that do not change status of users, passwords, or permissions. Detect an attempt to write to any file directly below “/” or “/root.” Identify when a database-server related program spawns a new process other than itself. Detect an attempt to change a program or thread’s namespace (commonly done as a part of creating a container) by calling setNS. Detect the initial process started in a privileged

container. Exceptions are made for known trusted images.

Detect the initial process started by a container that has a mount from a sensitive host directory. Exceptions are made for known trusted images.

Detect the initial process started by a container that is not in a list of allowed containers.

Detect any network activity performed by system binaries that are not expected to send or receive network traffic.

Detect an attempt to change users by calling setuid. Excluded: a) sudo/su and b) users “root” and “nobody” suing to itself.

Identifies when any files below “/dev” are created, with the exception of known programs that manage devices.

Detect when a Netcat Program runs inside a container that allows remote code execution.

Detect when network tools are launched inside of a container.

Detect when network tools are launched on a host.

Detect setuid or setgid bits set via chmod.

Detect symlink that has been created over sensitive files.

Detect new packet socket at the device driver (OSI Layer 2) level in a container.

Detect when stdout/stdin are redirected to a network connection in a container.

Detect when a new executable is created in a container due to chmod.

Detect when a new executable is created in a container due to open+create.

Detect an attempt to start a pod with a privileged container.

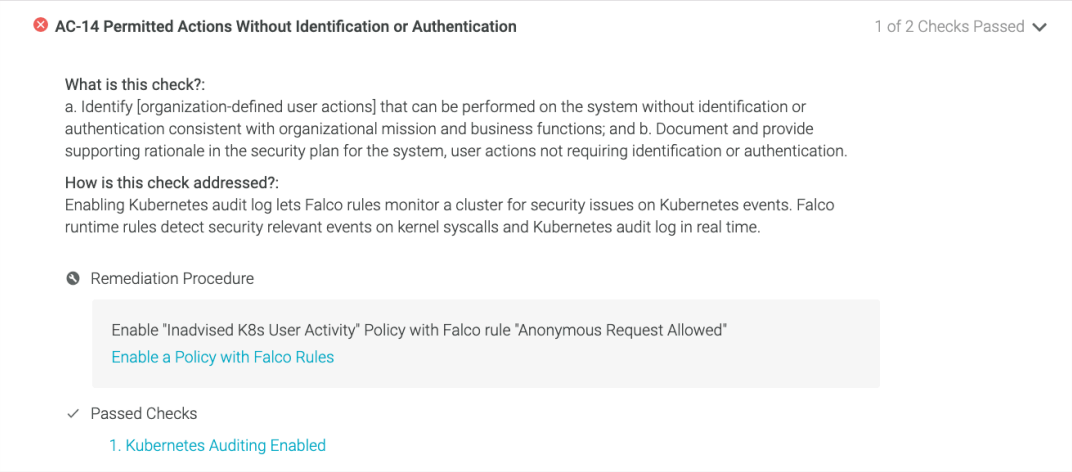
Detect an attempt to start a pod with a volume from a sensitive host directory.

Detect an attempt to create a pod in the kube-system or kube-public namespaces.

Control AC-14: Permitted Actions Without Identification or Authentication

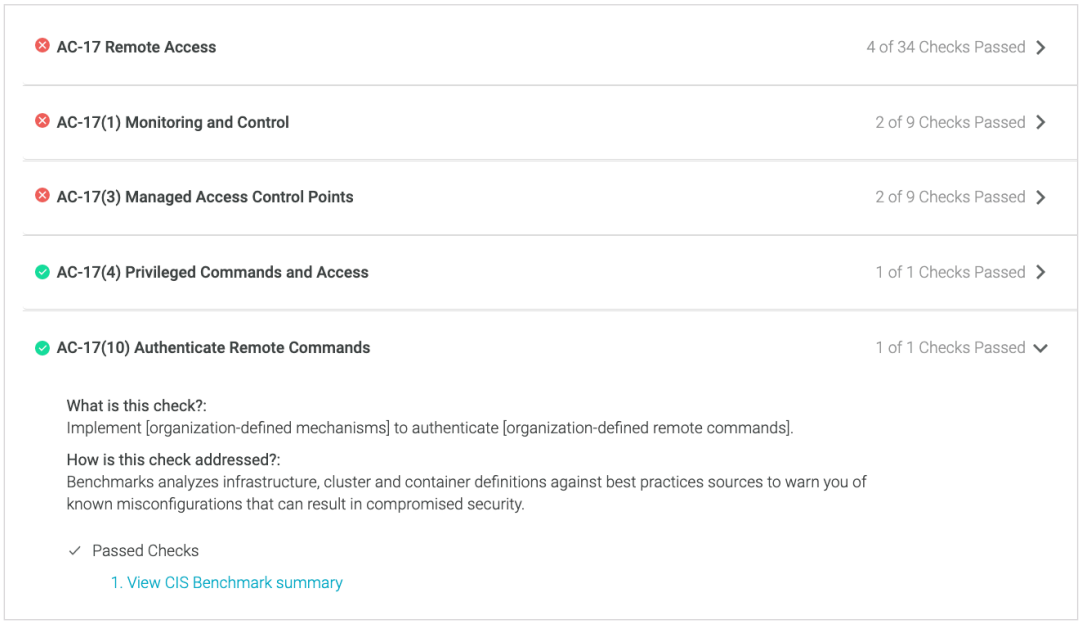
Provides a framework for identification of organization-defined actions that can be performed on the system without identification or authentication consistent with organizational mission and business functions.

Control ID	Control Name	How Sysdig helps
AC-14	Identify Permitted Actions Without Identification or Authorization	Detect requests made by allowed, anonymous users.



Control AC-17: Remote Access

Establishes usage restrictions, configuration/connection requirements, and implementation guidance for each type of remote access allowed.



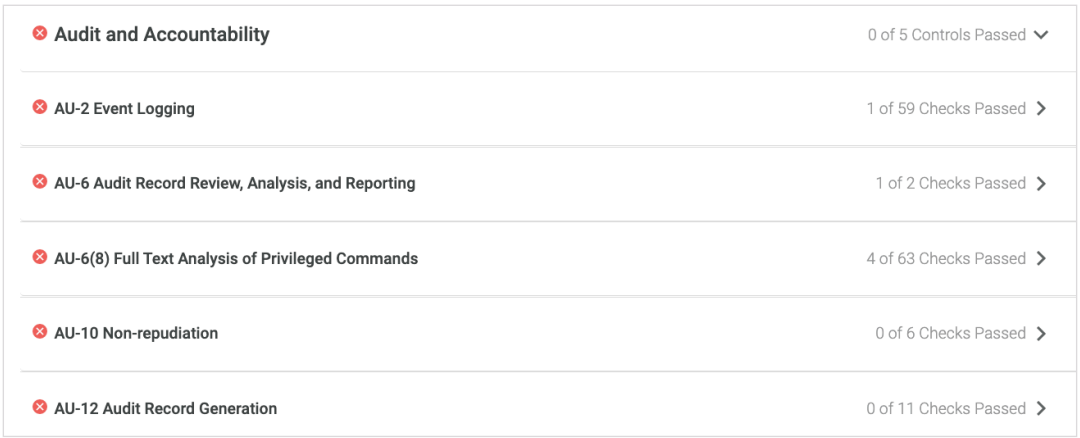
Control ID	Control Name	How Sysdig helps
AC-17	Remote Access Usage and Authorization	Identify when and where a terminal shell enters and exits a container. Detect attempts to attach to, or execute, a pod. Detect new SSH host connections not already connected to an allowed group of hosts. Specific applications monitored for attempts to spawn shells below non-shell applications. Detect any attempts to run an interactive command by a system (non logged-in) user. Detect Kubernetes operations by non-allowed users. Detect requests made by allowed, anonymous users. Identify when and where an attached terminal shell enters and exits a container. Detect any outbound connection to a destination outside of an allowed set of IPs, networks, or domain names.

		Detect any inbound connection from a source outside of an allowed set of IPs, networks, or domain names. Detect an attempt to run a program with a disallowed HTTP_PROXY environment variable. Detect inbound network activity performed by any interpreted program (i.e., Perl, Python, Ruby, and others). Detect outbound network activity performed by any interpreted program (i.e., Perl, Python, Ruby, and others). Detect unexpected traffic, specifically UDP traffic not on port 53 (DNS) or other commonly used ports. Detect attempts to contact the EC2 instance metadata service from a container. Detect attempts to contact the Cloud Instance Metadata Service from a container. Detect attempts to contact a Kubernetes API server from a container. Detect attempts to use Kubernetes NodePorts from a container. Detect when outbound connections are made to common miner pool ports. Detect evidence of cryptomining on Stratum protocol. Detect traffic to images outside the local subnet. Detect traffic that is not to authorized specific server processes and ports. Detect an attempt to start a pod using the host network. Detect an attempt to start a service with a NodePort service type. Detect creation of an ephemeral container.
AC-17 (1)	Monitoring and Control	Detect any outbound connection to a destination outside of an allowed set of IPs, networks, or domain names. Detect any inbound connection from a source outside of an allowed set of IPs, networks, or domain names. Detect an attempt to start a pod using the host network.

		Detect an attempt to start a service with a NodePort service type.
AC-17 (3)	Managed Access Control Point	Detect any outbound connection to a destination outside of an allowed set of IPs, networks, or domain names. Detect any inbound connection from a source outside of an allowed set of IPs, networks, or domain names. Detect an attempt to start a pod using the host network. Detect an attempt to start a service with a NodePort service type.

Control Group AU: Audit and Accountability

This group provides a framework for audit and accountability policies and procedures.



Control AU-2: Event Logging

Provides guidance on the types of events that the system is capable of logging in support of the audit function.

Control ID	Control Name	How Sysdig helps
AU-2	Event Logging	Detect attempts to attach to, or execute, a pod. Detect attempts to: • Create Kubernetes service accounts. • Delete Kubernetes service accounts. • Create Kubernetes clusters and/or roles. • Delete Kubernetes clusters and/or roles. Detect new SSH host connections not already connected to an allowed group of hosts. Specific applications monitored for attempts to spawn shells below non-shell applications. Detect any attempts to run an interactive command by a system (non logged-in) user. Detect any attempt to create a serviceaccount in the kube-system or kube-public namespaces. Detect any attempt to create a Role/ClusterRole that can perform write-related actions. Identify when and where an attached terminal shell enters and exits a container. Detect attempts to modify shell configuration files. Detect attempts to read shell configuration files by non-shell programs. Detect scheduled chron jobs. Detect attempts to write a file below a set of binary directories. Detect attempts to write a file below a monitored directory. Identify when a database-server related program spawns a new process other than itself.

Identifies when any files below “/dev” are created, with the exception of known programs that manage devices.

Detect when critical log files are cleared.

Detect processes running that clear bulk data from disk.

Detect hidden files or directories that have been created.

Detect when a Kubernetes client tool is executed inside a container.

Detect when CloudTrail logging has been disabled.

Detect an attempt to start a pod with a privileged container.

Detect an attempt to start a pod with a volume from a sensitive host directory.

Detect creation of an ephemeral container.

Detect an attempt to create a pod in the kube-system or kube-public namespaces.

Detect any attempt to create a deployment.

Detect any attempt to delete a deployment.

Detect any attempt to create a service.

Detect any attempt to delete a service.

Detect any attempt to create a configmap.

Detect any attempt to delete a configmap.

Detect any attempt to create a namespace.

Detect any attempt to delete a namespace.

Detect any attempt to create a secret (service account tokens are excluded).

Detect any attempt to delete a secret (service account tokens are excluded).

Match all Kubernetes audit events.

Detect an unsuccessful attempt to join a cluster for a node not in the list of allowed nodes.

Control AU-6: Audit Record Review, Analysis, and Reporting

Review and analysis of system audit records for indications of inappropriate or unusual activity.

Control ID	Control Name	How Sysdig helps
AU-6	Audit Record, Review, Analysis, and Reporting	Detect creation of an ephemeral container.
AU-6 (8)	Full Text Analysis of Privileged Commands	Identify when and where an attached terminal shell enters and exits a container. Detect new SSH host connections not already connected to an allowed group of hosts. Detect any attempt to create a ClusterRoleBinding to the cluster-admin user. Specific applications monitored for attempts to spawn shells below non-shell applications. Detect any attempts to run an interactive command by a system (non logged-in) user. Detect attempts to write a file below a set of binary directories. Detect attempts to write a file below a monitored directory. Detect an attempt to write to any file directly below "/" or "/root." Identify when a database-server related program spawns a new process other than itself. Detect an attempt to change a program or thread's namespace (commonly done as a part of creating a container) by calling setNS. Detect the initial process started in a privileged container. Exceptions are made for known trusted images. Detect the initial process started by a container that has a mount from a sensitive host directory. Exceptions are made for known trusted images.

Detect the initial process started by a container that is not in a list of allowed containers.

Detect any network activity performed by system binaries that are not expected to send or receive network traffic.

Detect an attempt to change users by calling setuid.

Identifies when any files below "/dev" are created, with the exception of known programs that manage devices.

Detect when a Netcat Program runs inside a container that allows remote code execution.

Detect when network tools are launched inside of a container.

Detect when network tools are launched on a host.

Detect when critical log files are cleared.

Detect processes running that clear bulk data from disk.

Detect shell history deletion.

Detect bash history deletion.

Detect symlink that has been created over sensitive files.

Detect when stdout/stdin are redirected to a network connection in a container.

Detect when a new executable is created in a container due to chmod.

Detect when a new executable is created in a container due to open+create.

Detect an attempt to start a pod with a container image that is not among a list of allowed images.

Detect an attempt to create a namespace outside of a set of known namespaces.

Detect when a node has successfully joined a cluster outside of the list of allowed nodes.

Control AU-8: Time Stamps

Use and record internal system clocks to generate timestamps for audit records.

Control ID	Control Name	How Sysdig helps
AU-8	Time Stamps	Detect when the blocking of public access to a bucket has been deleted.

Control AU-10: Non-Repudiation

Provide irrefutable evidence that an individual (or process acting on behalf of an individual) has performed specific actions that are covered by non-repudiation.

Control ID	Control Name	How Sysdig helps
AU-10	Non-Repudiation	Detect when critical log files are cleared. Detect processes running that clear bulk data from disk.

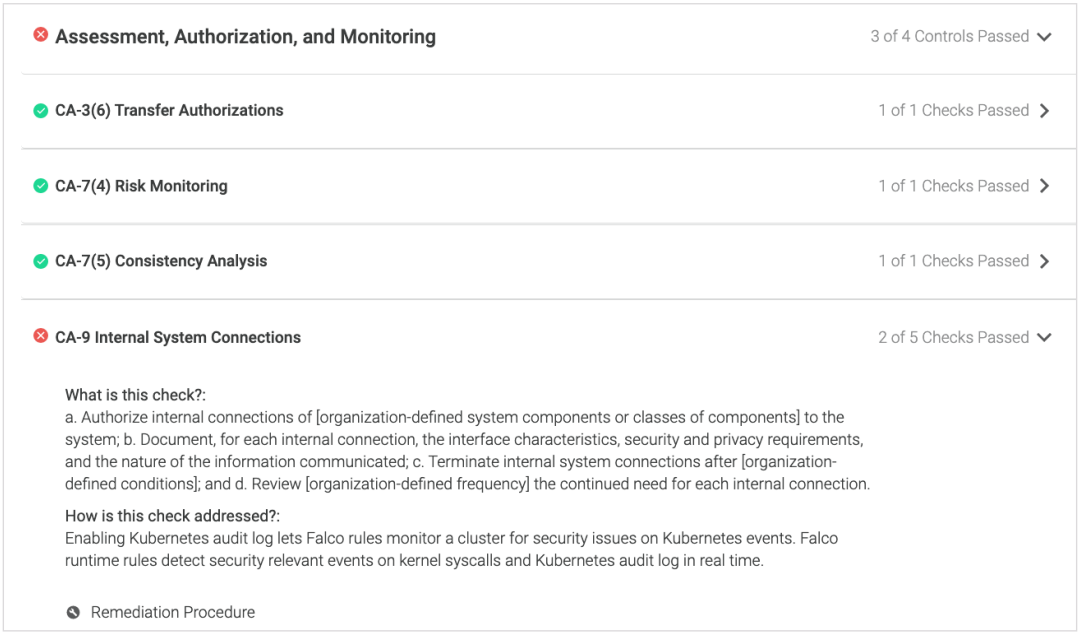
Control AU-12: Audit Record Generation

Provide audit record generation capability, including audit record for defined event types.

Control ID	Control Name	How Sysdig helps
AU-12	Audit Record Generation	Detect attempts to modify shell configuration files. Detect attempts to read shell configuration files by non-shell programs. Detect attempts to write a file below a set of binary directories. Detect attempts to write a file below a monitored directory.

Control Group CA: Assessment, Authorization, and Monitoring

Create assessment, authorization, and monitoring policies, designations for officials who can manage the policies, and reviews of those policies.



Control CA-2: Control Assessments

Guidance on user functionality, including user interface services, from system management functionality.

Control ID	Control Name	How Sysdig helps
CA-2 (2)	Specialized Assessments	Alerts when there are high severity packages with a known fix. Identifies when no npm packages are unknown to a VulnDB. Identifies when there are no high-severity non-OS packages. Identifies when there are no high-severity OS packages. Identifies when there are no Ruby gems unknown in a VulnDB. Identifies when there is not a general distribution image.

Control CA-3: Information Exchange

Approve and manage the exchange of information between systems, and document the interface characteristics, security requirements, controls, and responsibilities for each system.

Control ID	Control Name	How Sysdig helps
CA-3	Information Assessment	Instructs to block SSH port 22.

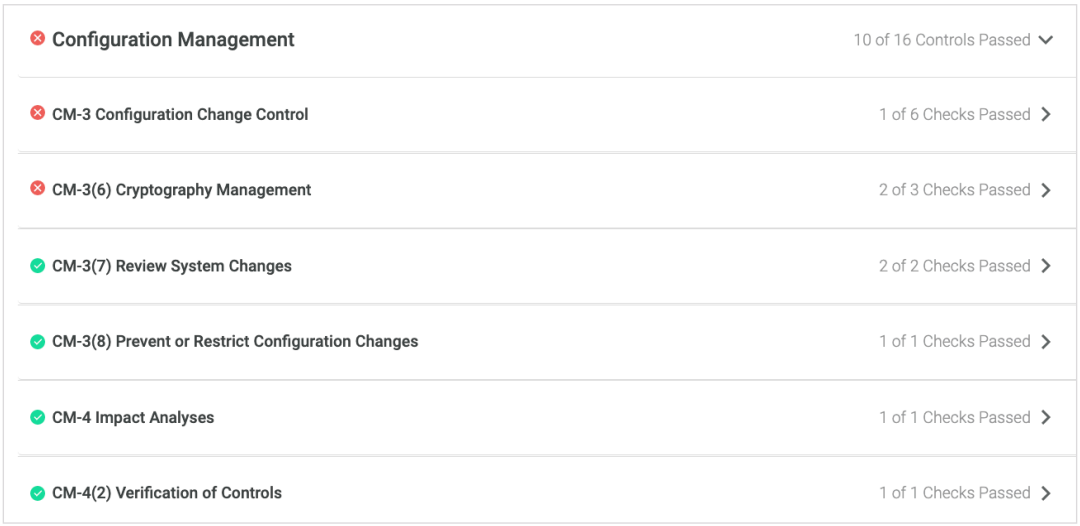
Control CA-9: Internal Systems Connections

Policies for authorization, documentation, review, and termination of internal connections.

Control ID	Control Name	How Sysdig helps
CA-9	Internal Systems Connections	Detect an attempt to read any sensitive file, such as those containing user, password, and authentication information. Exceptions are made for known trusted programs. Detect creation/modification of a configmap containing a private credential (such as AWS key or a password).

Control Group CM: Baseline Configuration

Provides baseline configurations for systems and system components, including connectivity, operational, and communications aspects of systems.



Control CM-2: Baseline Configuration

Develop a plan for baseline configuration of system components.

Control ID	Control Name	How Sysdig helps
CM-2	Baseline Configurations	Instructs to not use the latest label in Dockerfile. Instructs that you must not use insecure Dockerfile ADD commands. Instructs to not upgrade APT packages. Instructs to not upgrade YUM packages. Instructs to not upgrade APK packages.

Control CM-3: Configuration Change Control

Framework for reviewing and managing configuration-controlled changes to the system.

Control ID	Control Name	How Sysdig helps
CM-3	Configuration Change Control	Detect any attempt to create a Role/ClusterRole that can perform write-related actions. Detect an attempt to start a pod with a privileged container. Detect an attempt to start a pod with a volume from a sensitive host directory. Detect an attempt to create a pod in the kube-system or kube-public namespaces. Instructs to not use the latest label in Dockerfile. Instructs that you must not use insecure Dockerfile ADD commands. Instructs to not upgrade APT packages. Instructs to not upgrade YUM packages. Instructs to not upgrade APK packages.
CM-3 (6)	Cryptography Management	Detect when an ingress object without a TLS certificate has been created.

Control CM-5: Access Restrictions for Change

Define and enforce physical and logical access restrictions associated with changes to a system.

Control ID	Control Name	How Sysdig helps
CM-5	Access Restrictions for Change	Detect any attempt to create a serviceaccount in the kube-system or kube-public namespaces. Detect attempts to modify shell configuration files. Detect attempts to read shell configuration files by non-shell programs. Detect attempts to write a file below a set of binary directories.

Detect attempts to write a file below a monitored directory.

Detect an attempt to write to any file directly below "/" or "/root."

Detect an attempt to modify any file below a set of diary directories.

Detect an attempt to create a directory below a set of diary directories.

Detect package management processes running inside of a container.

Detect when critical log files are cleared.

Detect shell history deletion.

Detect bash history deletion.

Detect symlink that has been created over sensitive files.

Detect any attempt to create a deployment.

Detect any attempt to delete a deployment.

Detect any attempt to create a service.

Detect any attempt to delete a service.

Detect any attempt to create a configmap.

Detect any attempt to delete a configmap.

Detect any attempt to create a namespace.

Detect any attempt to delete a namespace.

Detect any attempt to create a secret (service account tokens are excluded).

Detect any attempt to delete a secret (service account tokens are excluded).

Control CM-7: Least Functionality

Configure a system to restrict use of specific functions, ports, protocols, software, and/or services.

Control ID	Control Name	How Sysdig helps
CM-7	Least Functionality	Detect an attempt to run a program with a disallowed HTTP_PROXY environment variable. Detect attempts to contact the EC2 instance metadata service from a container. Detect attempts to contact the Cloud Instance Metadata Service from a container. Detect evidence of cryptomining on Stratum protocol. Detect traffic that is not to authorized specific server processes and ports.
CM-7 (1)	Periodic Review	Detect inbound network activity performed by any interpreted program (i.e., Perl, Python, Ruby, and others). Detect when outbound connections are made to common miner pool ports. Detect evidence of cryptomining on Stratum protocol.
CM-7 (4)	Unauthorized Software: Deny-by-Exception	Detect the initial process started by a container that is not in a list of allowed containers.
CM-7 (6)	Confined Environments with Limited Privileges	Specific applications monitored for attempts to spawn shells below non-shell applications. Detect any attempts to run an interactive command by a system (non logged-in) user. Detect requests made by allowed, anonymous users. Detect any attempt to create a ClusterRoleBinding to the cluster-admin user. Detect any attempt to create a Role/ClusterRole with wildcard resources or verbs. Detect any attempt to create a Role/ClusterRole that can execute to pods.

	Detect Kubernetes operations by a user name that may be an admin with full access. Communicates that a user must be specified to run instead of default (root).

Control Group IA: Identification and Authentication

This group provides a framework for identification and authentication policies and procedures within a given system.

Control IA-3: Device Identification and Authentication

Guidance on user functionality, including user interface services, from system management functionality.

Control ID	Control Name	How Sysdig helps
IA-3	Device Identification and Authentication	Detect an unsuccessful attempt to join a cluster for a node not in the list of allowed nodes.

Control Group SC: System and Communications Protection

This group provides a framework for system-level communications policies and procedures.

Control SC-2: Audit Record Generation

Guidance on user functionality, including user interface services, from system management functionality.

Control ID	Control Name	How Sysdig helps
SC-2	Separation of System and User Functionality	Detect Kubernetes operations by non-allowed users.

Control SC-4: Information in Shared System Resources

Identification of unauthorized and unintended information transfer via shared system resources.

Control ID	Control Name	How Sysdig helps
SC-4	Information in Shared System Resources	Detect attempts to read files below SSH directories by non-SSH programs. Detect an attempt to read any sensitive file (i.e., files containing user/password/authentication information) by a trusted program after startup. Trusted programs might read these files at startup to load initial state, but not afterwards. Detect an attempt to read sensitive files by a trusted program after startup. This includes files that contain user, password, and/or authentication information. Detection is for trusted programs that would normally read these files at startup and load time, but not afterwards. Detect an attempt to read any sensitive file, such as those containing user, password, or authentication information. Exceptions are made for known trusted programs. Detect an attempt to change a program or thread's namespace (commonly done as a part of creating a container) by calling setNS. Detect activity for grep private keys and passwords.

		Detect remote file copy tools when launched in a container. Detect creation/modification of a configmap containing a private credential (such as AWS key or a password). Communicates to not include secrets in image contents. Instructs not to use secrets in image environment variables.

Control SC-5: Denial-of-Service Protection

Management of the effects and prevention of denial-of-service events.

Control ID	Control Name	How Sysdig helps
SC-5 (3)	Denial of Service	Communicates when a health check must be included.

Control SC-7: Boundary Protection

Framework for monitoring managed interfaces to, and within, systems and networks.

Control ID	Control Name	How Sysdig helps
SC-7	Boundary Protection	Detect an attempt to start a pod using the host network. Detect an attempt to start a service with a NodePort service type.
SC-7 (3)	Access Points	Detect any outbound connection to a destination outside of an allowed set of IPs, networks, or domain names. Detect any inbound connection from a source outside of an allowed set of IPs, networks, or domain names. Detect inbound network activity performed by any interpreted program (i.e., Perl, Python, Ruby, and others).

		Detect outbound network activity performed by any interpreted program (i.e., Perl, Python, Ruby, and others). Detect when outbound connections are made to common miner pool ports. Detect evidence of cryptomining on Stratum protocol. Detect an attempt to start a pod using the host network. Detect an attempt to start a service with a NodePort service type.
SC-7 (10)	Prevent Exfiltration	Detect inbound network activity performed by any interpreted program (i.e., Perl, Python, Ruby, and others). Detect outbound network activity performed by any interpreted program (i.e., Perl, Python, Ruby, and others).

Control SC-8: Transmission Confidentiality and Integrity

Management, protection, and integrity of transmitted information.

Control ID	Control Name	How Sysdig helps
SC-8	Transmission Confidentiality and Integrity	Detect when an ingress object without a TLS certificate has been created.
SC-8 (1)	Cryptographic Protection	Detect creation of an HTTP target group that is not using SSL.

Control SC-12: Cryptographic Key Establishment and Management

Establishing and managing cryptographic keys when cryptography is used.

Control ID	Control Name	How Sysdig helps
SC-12 (3)	Asymmetric Keys	Detect when an ingress object without a TLS certificate has been created.

Control SC-17: Public Key Infrastructure Capabilities

Issuing and/or obtaining public key certificates from approved service providers.

Control ID	Control Name	How Sysdig helps
SC-12(3)	Asymmetric Keys	Detect when an ingress object without a TLS certificate has been created.

Control SC-28: Cryptographic Protection

Implementation of cryptographic mechanisms to provide unauthorized disclosure and modification of organization-defined information at rest.

Control ID	Control Name	How Sysdig helps
SC-28 (1)	Cryptographic Protection	Communicates to not include secrets in image contents. Instructs not to use secrets in image environment variables.

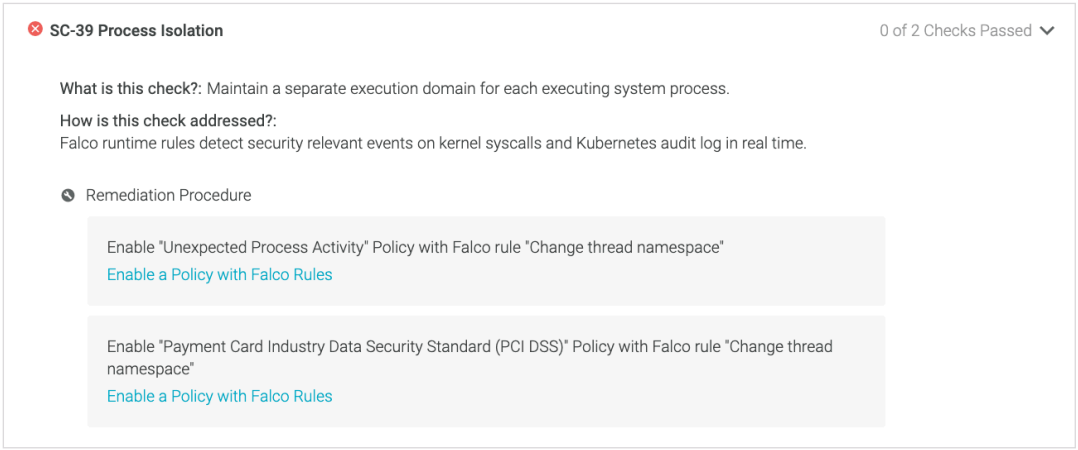
Control SC-30: Concealment and Misdirection

Guides how to employ concealment and misdirection techniques.

Control ID	Control Name	How Sysdig helps
SC-30	Concealment and Misdirection	Communicates to not include secrets in image contents.
SC-30 (5)	Concealment of System Components	Instructs not to use secrets in image environment variables.

Control SC-39: Process Isolation

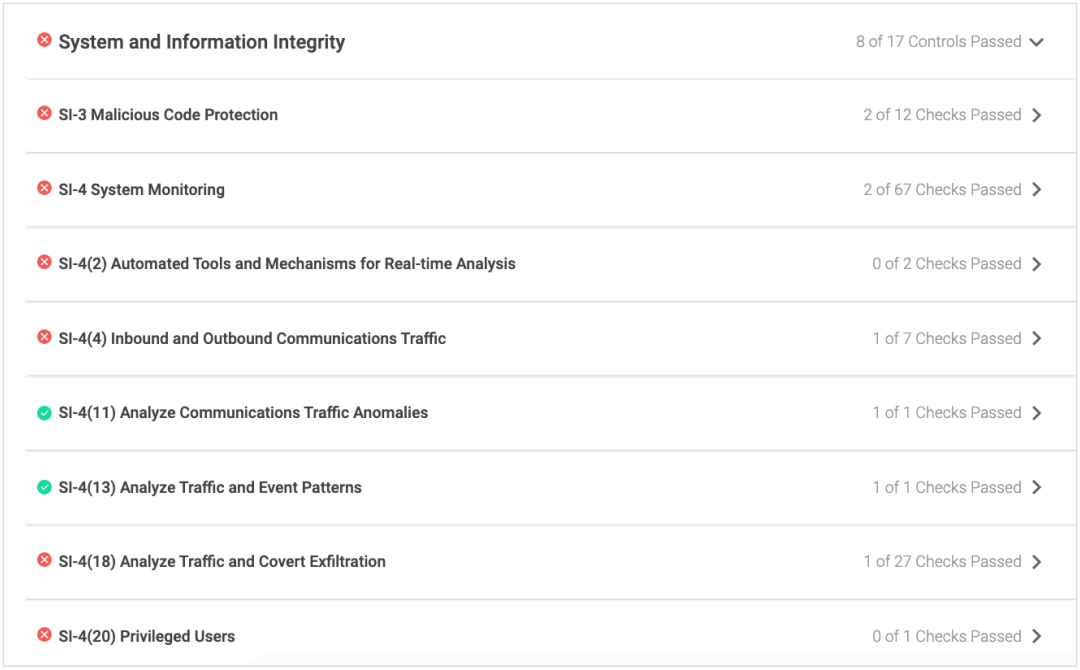
Maintenance of a separate execution domain for each executing system process.



Control ID	Control Name	How Sysdig helps
SC-39	Process Isolation	Detect an attempt to change a program or thread's namespace (commonly done as a part of creating a container) by calling setNS.

Control Group SI: System and Information Integrity

This group provides a framework for system integrity policies and procedures.



Control SI-3: Malicious Code Protection

Maintenance of a separate execution domain for each executing system process.

Control ID	Control Name	How Sysdig helps
SI-3	Malicious Code Protection	Specific applications monitored for attempts to spawn shells below non-shell applications. Detect any attempts to run an interactive command by a system (non logged-in) user. Detect when package repositories get updated. Detect package management processes running inside of a container. Detect when a Netcat Program runs inside a container that allows remote code execution. Detect when network tools are launched inside of a container. Detect when network tools are launched on a host.

		Detect symlink that has been created over sensitive files. Alerts when there are high severity packages with a known fix. Identifies when no npm packages are unknown to a VulnDB. Identifies when there are no high-severity non-OS packages. Identifies when there are no high-severity OS packages. Identifies when there are no Ruby gems unknown in a VulnDB. Identifies when there is not a general distribution image.
SI-3 (2)	Automatic Updates	Ensures that VulnDB is recent.

Control SI-4: System Monitoring

Identification and monitoring of attacks and indicators of potential attacks.

Control ID	Control Name	How Sysdig helps
SI-4	System Monitoring	Detect new SSH host connections not already connected to an allowed group of hosts. Specific applications monitored for attempts to spawn shells below non-shell applications. Detect any attempts to run an interactive command by a system (non logged-in) user. Detect any attempt to create a Role/ClusterRole that can perform write-related actions. Detect any outbound connection to a destination outside of an allowed set of IPs, networks, or domain names. Detect any inbound connection from a source outside of an allowed set of IPs, networks, or domain names. Detect scheduled chron jobs. Detect when package repositories get updated.

Detect attempts to write a file below a set of binary directories.

Detect attempts to write a file below a monitored directory.

Detect attempts to write any file below/etc.

Detect an attempt to write to the RPM database by a non RPM-related program.

Identify when a database-server related program spawns a new process other than itself.

Detect an attempt to modify any file below a set of diary directories.

Detect an attempt to create a directory below a set of diary directories.

Detect an attempt to run a program with a disallowed HTTP_PROXY environment variable.

Detect unexpected traffic, specifically, UDP traffic not on port 53 (DNS) or other commonly used ports.

Identifies when any files below "/dev" are created, with the exception of known programs that manage devices.

Detect attempts to contact the EC2 instance metadata service from a container.

Detect attempts to contact the Cloud Instance Metadata Service from a container.

Detect attempts to contact a Kubernetes API server from a container.

Detect attempts to use Kubernetes NodePorts from a container.

Detect package management processes running inside of a container.

Detect when a Netcat Program runs inside a container that allows remote code execution.

Detect when network tools are launched inside of a container.

Detect when network tools are launched on a host.

Detect setuid or setgid bits set via chmod.

Detect when a Kubernetes client tool is executed inside a container.

		Detect traffic to images outside the local subnet. Detect traffic that is not to authorized specific server processes and ports. Detect an update of an existing CloudTrail trail. Detect deletion of CloudWatch alerts. Detect deletion of a CloudWatch log group. Detect deletion of a CloudWatch log stream. Detect an attempt to start a pod with a privileged container. Detect an attempt to start a pod with a volume from a sensitive host directory. Detect an attempt to start a pod using the host network. Detect an attempt to start a service with a NodePort service type. Detect an attempt to create a pod in the kube-system or kube-public namespaces.
SI-4 (2)	Automated Tools and Mechanisms for Real-Time Analysis	Specific applications monitored for attempts to spawn shells below non-shell applications. Detect any attempts to run an interactive command by a system (non logged-in) user.
SI-4 (4)	Inbound and Outbound Communications Traffic	Detect any outbound connection to a destination outside of an allowed set of IPs, networks, or domain names. Detect any inbound connection from a source outside of an allowed set of IPs, networks, or domain names. Detect when outbound connections are made to common miner pool ports.
SI-4 (18)	Analyze Traffic and Covert Exfiltration	Detect any outbound connection to a destination outside of an allowed set of IPs, networks, or domain names. Detect any inbound connection from a source outside of an allowed set of IPs, networks, or domain names. Detect attempts to read files below SSH directories by non-SSH programs. Detect an attempt to read sensitive files by a trusted program after startup. This includes files that contain

		user, password, and/or authentication information. Detection is for trusted programs that would normally read these files at startup and load time, but not afterwards. Detect an attempt to read any sensitive file, such as those containing user, password, or authentication information. Exceptions are made for known trusted programs. Detect an attempt to run a program with a disallowed HTTP_PROXY environment variable. Detect unexpected traffic, specifically, UDP traffic not on port 53 (DNS) or other commonly used ports. Detect attempts to contact the EC2 instance metadata service from a container. Detect attempts to contact the Cloud Instance Metadata Service from a container. Detect attempts to contact a Kubernetes API server from a container. Detect attempts to use Kubernetes NodePorts from a container. Detect activity for grep private keys and passwords. Detect remote file copy tools when launched in a container. Detect traffic to images outside the local subnet. Detect traffic that is not to authorized specific server processes and ports.
SI-4 (20)	Privileged Users	Detect setuid or setgid bits set via chmod.
SI-4 (22)	Unauthorized Network Services	Detect setuid or setgid bits set via chmod.
SI-4 (24)	Indicators of Compromise	Specific applications monitored for attempts to spawn shells below non-shell applications. Detect any attempts to run an interactive command by a system (non logged-in) user.

Control SI-7: Software, Firmware, and Information Integrity

Employ integrity verification tools to detect unauthorized changes to software, firmware, and information.

Control ID	Control Name	How Sysdig helps
SI-7	Software, Firmware, and Information Integrity	Detect attempts to modify shell configuration files. Detect attempts to read shell configuration files by non-shell programs. Detect when package repositories get updated. Detect attempts to write a file below a set of binary directories. Detect attempts to write a file below a monitored directory. Detect attempts to write any file below/etc. Detect an attempt to write to the RPM database by a non RPM-related program. Detect an attempt to modify any file below a set of diary directories. Detect an attempt to create a directory below a set of diary directories. Detect any network activity performed by system binaries that are not expected to send or receive network traffic. Detect package management processes running inside of a container. Detect shell history deletion. Detect bash history deletion. Detect setuid or setgid bits set via chmod. Detect hidden files or directories that have been created.

To learn more about how Sysdig Secure validates compliance visit
https://sysdig.com/products/kubernetes-security/container-compliance/

You can also sign-up for a Sysdig Secure free 30-day trial at
https://sysdig.com/company/free-trial/

www.sysdig.com