**Watch Tower Cybersecurity Framework.**

**A Guided Process in Securing Critical IT Systems from Internal Threats.**

**v1.0**

Contents

[Scope: iii](#_Toc177821437)

Scope:  
This framework is designed to secure **critical systems handling sensitive data** from **internal threats**, such as insider misuse, misconfigurations, and design flaws. These threats come from trusted users within the organization who may have legitimate access but pose a risk, either intentionally or accidentally.

WatchTower is tailored to focus on the challenges of **internal cybersecurity** by refactoring the system architecture, enhancing monitoring, and using advanced internal threat detection technologies like **canary files, tripwires, honeypots**, and similar solutions.

# Introduction

Internal threats are inherently dangerous because they originate within the organization’s trusted perimeter. Unlike external threats, insider attacks can be harder to detect and often exploit known system vulnerabilities, misconfigurations, and access privileges. These threats can stem from employees, contractors, or third-party vendors who misuse their legitimate access to the system or accidentally expose sensitive data.

The WatchTower Framework emphasizes preventing, detecting, and mitigating internal cybersecurity threats. It focuses on building resilient systems by integrating internal threat detection technologies, refining system design, and establishing processes that mitigate risks posed by trusted insiders.

# Key Components of the WatchTower Framework

## Risk Assessment and Threat Modelling

### **Data Sensitivity and Threat Classification**

* **Classify Sensitive Data**: Identify critical systems that handle sensitive data, categorizing it based on business impact (e.g., financial records, personal health data, intellectual property).
* **Internal Threat Risk Modelling**: Use **internal threat modelling** to simulate how trusted users or insiders might exploit vulnerabilities. Focus on **misconfigurations**, **privilege misuse**, and **accidental breaches**.

### ****System**** Vulnerability Assessment

* Conduct detailed **system vulnerability assessments**, focusing on how insiders could exploit existing vulnerabilities such as **API misconfigurations**, weak access controls, or improper data handling.
* Assess internal processes for security gaps in **configuration management**, **privilege escalation**, and **data governance**.

### ****Proactive Threat Identification****

* Deploy proactive strategies that anticipate how insiders might misuse system components, exploiting design flaws or weak configurations. This includes monitoring for common insider threat vectors such as:
  + **Privileged user misuse**
  + **Unauthorized data access**
  + **Data exfiltration via legitimate access paths**

## Refactoring System Architecture for Insider Threats

### Zero Trust Architecture

* + Implement **Zero Trust** principles by ensuring that no user, inside or outside the network, is inherently trusted. Every request for access must be authenticated and verified, even from internal sources. This can leverage **FISMA** guidelines on secure network design and **NERC-CIP**'s approaches to critical infrastructure protection.

### Micro-Segmentation:

* + Divide the internal network into **smaller, isolated segments** to limit lateral movement. This is particularly useful in critical systems where the exposure of one subsystem should not compromise the entire network. Each segment should have its own security policy.

### Access Control Refinement:

* + Implement **Role-Based Access Control (RBAC)** and **Just-in-Time (JIT) access**, reducing the scope and duration of elevated privileges. **ISO 27002** provides detailed guidelines on managing access control effectively. Ensure that **privileged access** is continuously monitored, and **least privilege** is enforced throughout.

### Deception Technologies (Canaries, Honeypots):

* + Introduce deception techniques such as **canary files** and **honeypots** within critical systems to detect insider threats. These decoys alert security teams to potential malicious activity, ensuring quick response times. This concept is supported by **NIST’s Detect Function** in the cybersecurity framework.

## Advanced internal Threat Detection Technologies

### Canary Files and Decoy Systems

* Deploy **canary files** in critical systems as decoys, designed to attract malicious insiders or those with unauthorized access. These files mimic sensitive data but are rigged to trigger an alert if accessed or modified.
* **Customize canary files** to reflect critical assets in your system, ensuring that their misuse generates high-priority alerts.
* Place these files in **strategic locations** such as shared drives, databases, and endpoints with high-value data.

### Honeypots

* Set up **honeypots** within the network to mimic systems or databases that house sensitive data. Honeypots are used to lure insiders attempting to explore or compromise data beyond their access levels.
* Honeypots can serve as **early-warning systems**, alerting the security team to any suspicious or unauthorized attempts to interact with seemingly valuable resources.

### Tripwires and File Integrity Monitoring (FIM)

* Implement **tripwire systems** that monitor key files and configuration settings for unauthorized changes. When a tripwire detects any changes, such as modification of sensitive files or system configuration, it triggers an immediate alert.
* Use **File Integrity Monitoring (FIM)** tools to regularly compare system files against a secure baseline. Any discrepancies indicate possible insider tampering, prompting further investigation.

### Behavioral Analytics and Internal Monitoring

* Use **User and Entity Behavior Analytics (UEBA)** to detect anomalies in user behavior that could indicate insider threats. The system should monitor baseline behaviors, such as login times, access patterns, and data handling activities, and alert when deviations occur.
* Behavioral monitoring tools should be tuned to detect suspicious behavior like:
  + Unusual access to sensitive data.
  + Attempts to bypass standard workflows or permissions.
  + Large data transfers or downloads.

## Access and Identity Management

### 4.1 Strong Multi-Factor Authentication (MFA)

* Enforce **Multi-Factor Authentication (MFA)** for all access to critical systems and sensitive data. MFA should be required not only for external access but for **internal system access**, ensuring that even insiders face an additional authentication layer.

### 4.2 Least Privilege and Privilege Management

* Implement **Least Privilege** principles, ensuring that each user has the minimum access required to perform their role. Regularly review access privileges to identify and remove unnecessary rights.
* Use **Privileged Access Management (PAM)** to control and monitor the use of privileged accounts. This includes **just-in-time (JIT) access** for elevated permissions, where access is granted only when needed and revoked immediately afterward.

### 4.3 Contextual Access Control

* Introduce **contextual access control** based on factors such as time of day, location, and the device used. If an insider attempts to access sensitive data under unusual circumstances, alerts should be triggered, or access should be denied.
* Implement **role-based access control (RBAC)** to manage permissions based on predefined roles within the organization, ensuring consistency and security.

## Incident Detection, Response, and Recovery

### Internal Threat Detection and Response

* Build a dedicated insider threat detection system that automatically flags unusual behavior, such as large-scale data access, copying sensitive files, or attempts to bypass security controls.
* Create response playbooks for common insider threat scenarios, such as data exfiltration or unauthorized privilege escalation, ensuring the response is swift and decisive.

### Automated Incident Response

* Enable automated incident response for high-risk behaviors identified by monitoring tools. For example, if an insider accesses a canary file, the system should immediately isolate the user’s account and initiate an investigation.
* For more serious breaches, trigger automated processes that include revoking access rights, quarantining affected systems, and preserving forensic evidence.

### Post-Incident Recovery and Improvement

* After responding to an insider threat, ensure rapid data recovery and restoration of services using backups and disaster recovery plans.
* Conduct post-incident reviews to identify weaknesses in the system, and apply those insights to continuously improve the system’s resilience to insider threats.

## Continuous Monitoring, Audits, and Training

### Real-Time Continuous Monitoring

* Continuously monitor all systems handling sensitive data using internal threat detection technologies like **canaries, honeypots, and FIM**. All activity should be logged and analyzed in real time, with automated alerts set for any suspicious or anomalous behavior.

### Regular Audits and System Reviews

* Conduct frequent audits focused on **internal system security**. These audits should review access logs, privilege changes, and system configuration updates to ensure there are no hidden vulnerabilities or misconfigurations that insiders could exploit.
* Schedule periodic **vulnerability scans** for both system components and internal networks.

### Insider Threat Awareness Training

* Implement **awareness programs** that train employees to recognize potential insider threats and report suspicious behaviour. Continuous education helps prevent accidental insider threats and fosters a culture of security consciousness.
* Specialized training for employees with **elevated privileges** or those with access to sensitive data is critical, ensuring they understand the unique risks and responsibilities they carry.