

The background features a large, dark blue mountain peak on the left side. Several stylized, layered clouds in shades of blue and green are scattered across the image. A white line outlines the right side of the mountain peak. The overall color palette is dominated by various shades of blue and green.

CADO//

# Ultimate Guide to Incident Response in Azure

# Table of Contents



3	Introduction
4	Key Azure Log Sources
8	Before the Incident: Proactive Security Measures
11	Service-Specific Incident Response Strategies
16	Automating Incident Response
19	Forensic Analysis in Azure
22	Post-Incident Review and Continuous Improvement
23	Common Challenges in Azure Incident Response
25	Tools and Resources
28	Conclusion and Recommendations
30	What is the Cado Platform?



# Introduction

Investigating and responding to incidents in cloud environments like Azure is fundamentally different to on-premise. Further, without the right tools and processes in place, it can be more complicated. There are over 200 products and services in Azure, each with different security best practices and data sources. While the cloud can make incident response more complex, it also enables some fantastic possibilities. For example, by leveraging cloud resources to collect, process, and store evidence, you can expedite the end-to-end incident response process in ways that would be unthinkable on-premise.

Azure's incident response advice mentions two critical components to consider when measuring how well your organization is prepared to reduce risk: mean time to acknowledge (MTTA) and mean time to respond (MTTR). The best practices outlined in this playbook were crafted with these two key metrics in mind with the goal of yielding noticeable improvement in both.



This guide aims to provide a comprehensive overview of incident response in Azure, equipping security teams with the knowledge and tools necessary to effectively handle security incidents in the cloud.



# Key Azure Log Sources

Incident response in **Microsoft Azure** relies heavily on the ability to collect, analyze, and act on log data. Azure provides multiple logging services, each serving different purposes, from tracking administrative changes to monitoring network activity. Understanding these log sources and their capabilities is essential for effective detection, investigation, and response to security incidents.

Below are the key Azure log sources that security teams should prioritize:

## Azure Activity Logs

Azure **Activity Logs** provide a record of subscription-level administrative actions, such as creating virtual machines, modifying network configurations, or changing security settings. These logs are essential for auditing user actions and monitoring unauthorized changes in an Azure environment.

### BEST PRACTICES INCLUDE:

- Enabling **Activity Logs** across all subscriptions.
- Forwarding logs to **Azure Monitor Logs** for centralized analysis.
- Configuring alerts for high-risk administrative actions, such as role assignments or resource deletions.

## Azure Resource Logs

Azure **Resource Logs** (formerly known as Diagnostic Logs) capture **data plane** operations for specific Azure services, such as Virtual Machines (VMs), Key Vaults, and Storage Accounts. These logs provide insights into how services are used and help detect anomalous activity.

### RECOMMENDATIONS:

- Enable **Resource Logs** for all critical Azure resources.
- Store logs in **Azure Log Analytics** for correlation and threat analysis.
- Use **Microsoft Sentinel** to detect suspicious patterns in log data.

## Azure Active Directory (Azure AD) Logs

Azure **AD Logs** record authentication events, identity changes, and security-related activities. These logs are crucial for detecting unauthorized access attempts, password spraying, and other identity-based threats.

### KEY LOG TYPES INCLUDE:

- **Sign-in Logs:** Tracks user and service authentication attempts.
- **Audit Logs:** Logs configuration changes and administrative actions.
- **Risky Sign-ins:** Identifies suspicious login patterns, such as impossible travel or brute-force attempts.

### BEST PRACTICES:

- Enable **Conditional Access Policies** to enforce security baselines.
- Use **Azure AD Identity Protection** to automate threat detection.
- Forward logs to **Microsoft Sentinel** for real-time monitoring.



## Azure Monitor & Virtual Machine Logs

Azure **Monitor Logs** provide centralized visibility across applications, infrastructure, and network activity. They aggregate logs from various Azure services, including Virtual Machines (VMs), to help security teams analyze potential threats.

### FOR WINDOWS AND LINUX VMS:

- Use **Azure Diagnostics Extension** to collect system logs.
- Configure **Log Analytics Agent** to send logs to **Azure Monitor**.
- Analyze event logs to detect signs of system compromise, unauthorized access, or malware execution.

## Application Insights & Application Logs

**Application Insights** monitors the performance and health of web applications hosted in Azure. It captures telemetry data, including request rates, response times, and custom logs.

**Application Logs** capture internal application errors, warnings, and debugging information. These logs help security teams detect application-layer attacks, such as SQL injection or API abuse.

### BEST PRACTICES:

- Enable **Application Insights** for all mission-critical applications.
- Correlate logs with **Azure Security Center** alerts for enhanced threat detection.
- Implement **Web Application Firewall (WAF)** Logs to detect and block malicious HTTP requests.



## Azure Storage Analytics Logs

Azure **Storage Analytics Logs** provide detailed insights into storage account activity, including access requests, authentication attempts, and data transfers.

### USE CASES:

- Detecting unauthorized access to **Azure Blob Storage**.
- Monitoring **SAS Token** usage to prevent excessive data exfiltration.
- Tracking failed access attempts to identify potential brute-force attacks.

### BEST PRACTICES:

- Enable logging for all storage accounts containing sensitive data.
- Analyze logs for **unusual download patterns** or **high request volumes**.
- Store logs in **Azure Log Analytics** for correlation with other security events.

## Azure Network Security Group (NSG) Flow Logs

Azure **NSG Flow Logs** provide network-level visibility by capturing accepted and denied traffic within Virtual Networks (VNETs). These logs help detect unauthorized traffic flows, lateral movement, and potential data exfiltration attempts.

### BEST PRACTICES INCLUDE:

- Enabling **NSG Flow Logs** in all critical subnets.
- Forwarding logs to **Azure Log Analytics** for deeper analysis.
- Using **Microsoft Sentinel** to detect anomalous network behavior.



## Microsoft Defender for Cloud Security Alerts

**Microsoft Defender for Cloud** provides real-time security alerts for potential threats in an Azure environment. These alerts include detections from various security tools, such as:

- **Azure Defender for Servers** (for VM-based threats).
- **Azure Defender for Storage** (for data exfiltration monitoring).
- **Azure Defender for Key Vault** (for unauthorized access detection).

### BEST PRACTICES INCLUDE:

- Integrate Defender for Cloud with **Microsoft Sentinel** for automated incident response.
- Enable **Just-in-Time VM Access** to reduce exposure to brute-force attacks.
- Automate remediation using **Azure Logic Apps** and **Playbooks**.

## Other Azure Logging Sources

Azure provides additional security-focused logging capabilities, including:

- **Azure Web Application Firewall (WAF) Logs:** Detects and blocks web-based attacks.
- **Azure Firewall Logs:** Captures network traffic logs for firewall rules.
- **Azure SQL Auditing Logs:** Tracks database access and potential SQL injection attempts.

# Before the Incident: Proactive Security Measures

The following best practices can help security teams reduce the likelihood that an incident will occur, and in the event that it does, drastically decrease recovery time.



## Understand and Protect Critical Data

Identify and classify sensitive data, such as **Personally Identifiable Information (PII)** and **Payment Card Industry (PCI)** data, using **Azure Information Protection (AIP)**. Implement **role-based access control (RBAC)** and enforce least privilege for sensitive data access.

For more information: [Azure Information Protection Documentation](#)



## Backup and Recovery Planning

Enable **Azure Backup** for critical workloads and store backups in **immutable storage** to prevent tampering. Regularly test restores to ensure reliability. Implement **Geo-Redundant Storage (GRS)** for disaster recovery.

For more information: [Azure Backup Documentation](#)



## Secure Administrative Access

Enforce **Just-In-Time (JIT)** access with **Azure Privileged Identity Management (PIM)**, disable **legacy authentication**, and require **Multi-Factor Authentication (MFA)** for all accounts. Monitor risky sign-ins and audit privileged actions.

For more information: [Best Practices for Entra ID Roles](#)

## Network and Remote Access Controls

Restrict exposure of Azure **Virtual Machines (VMs)** by enabling **Network Security Groups (NSGs)**, **Just-In-Time (JIT) access**, and **Azure Bastion** for secure remote access. Avoid exposing **RDP** and **SSH** to the public internet.

For more information: [Just-In-Time VM Access](#)

## Enable Logging and Monitoring

Ensure **forensic readiness** by enabling logs for key security events:

- **Activity Logs:** Track subscription-level management events.
- **Resource Logs:** Capture access and operational events.
- **Azure AD Logs:** Monitor authentication and identity risks.
- **NSG Flow Logs:** Record inbound/outbound network traffic.
- **Microsoft Defender for Cloud Alerts:** Detect and respond to threats.

Both [Data Dog](#) and [Secure Works](#) have great tutorials on how to ensure full logging is enabled.

For more information: [Azure Monitor Data Collection](#)



## Incident Readiness and Response Planning

Regularly conduct **tabletop exercises** to simulate incidents and refine response strategies. Define escalation protocols, external engagement policies, and secure communication methods in case primary systems are compromised.

Executives should be prepared to answer to the following questions in advance of any incident:

- Under what circumstances do you notify law enforcement, regulatory authorities, auditors, and the board?
- Will we pay a ransom? If so, how?
- If required, which out-sourced incident response firm will you work with?
- If you lose access to core IT systems for an extended period of time?
- Do you have business continuity and disaster recovery plans in place?
- If the primary communication methods are either unavailable or compromised, do you have backup or out-of-band communications available?
- What working hours are incident responders expected to work in a high severity Incident?
- Do you have access to the data required to perform an investigation in all products and services?

For more information: [Microsoft Incident Response Plan Guide](#)



# Service-Specific Incident Response Strategies

Incident response in Azure varies depending on the affected services. Security teams need to apply tailored approaches when investigating incidents involving **Active Directory (AD)**, **Virtual Machines (VMs)**, and **Azure Kubernetes Service (AKS)**. This section provides **detailed guidance on detecting, containing, and responding to security incidents** in these critical Azure environments.

## Active Directory Incident Response (Now Entra ID)

Azure Entra ID is a key identity and access management service that controls authentication and authorization for users, applications, and services. Threat actors frequently target **Entra ID** to gain unauthorized access or escalate privileges.

### IDENTIFYING AND DISABLING LEGACY AUTHENTICATION

**Legacy authentication methods** (such as **Basic Authentication for Exchange Online**) are a common attack vector, allowing **password spray** and **credential stuffing** attacks.

#### Steps to Identify and Block Legacy Authentication:

1. **Navigate to the Azure Portal:**
  - Go to **Entra ID → Sign-in Logs**.
  - Click on **Columns → Client App**, if not already displayed.
  - Select **Add Filters → Client App →** Check all legacy authentication methods.
2. **Block Legacy Authentication using Conditional Access Policies:**
  - Navigate to **Entra ID → Security → Conditional Access**.
  - Create a new policy:
    - **Include:** All users.
    - **Exclude:** Service accounts (if necessary).
    - **Conditions:** Sign-in risk level = Medium or High.
    - **Grant:** Block access.
3. **Monitor for attempts to bypass blocks using Security Reports:**
  - **Security → Reports → Sign-in Logs**
  - Review blocked authentication attempts and investigate suspicious IPs.

For more information: [Microsoft's Guide on Blocking Legacy Authentication](#)

## REVIEWING RISKY SIGN-INS IN ENTRA ID

Azure AD provides built-in **risk-based detection** to identify potentially compromised accounts.

### Steps to Investigate Risky Sign-ins:

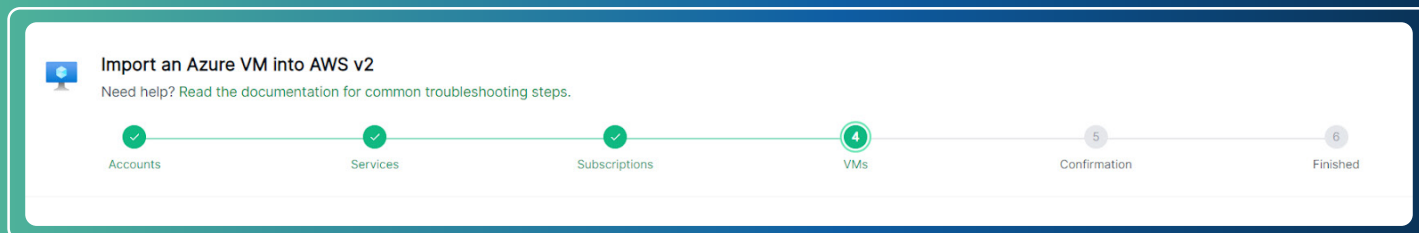
1. **Navigate to Entra ID → Security → Risky Sign-ins.**
  - Look for anomalies such as **impossible travel activity** or **sign-ins from anonymous IP addresses**.
2. **Check the “Risk Detections” tab:**
  - Provides reports for **password spray attacks**, **leaked credentials**, and **high-risk login attempts**.
  - Data retention: **90 days**.
3. **Remediate Risky Sign-ins:**
4. **Force a password reset for affected accounts.**
5. **Enable Multi-Factor Authentication (MFA) if not already enforced.**
6. **Block high-risk users using the following command:**

```
az ad user update --id <user_id> --account-enabled false
```

For more information: [Microsoft Risk-Based Sign-in Analysis](#)

## Virtual Machines (VMs) Incident Response

Azure Virtual Machines (**Azure VMs**) are frequently **targeted by attackers** who aim to escalate privileges, exfiltrate sensitive data, or deploy **ransomware**. Responding to **VM incidents** requires **snapshotting compromised instances**, collecting forensic evidence, and implementing **containment strategies**.



The Cado Platform can automatically acquire Azure Compute instances for investigation.



# Azure Kubernetes Service (AKS)

**AKS** is a **managed Kubernetes service** that allows enterprises to run containerized workloads. Attackers frequently **target Kubernetes clusters** to **exploit misconfigurations, escalate privileges, or exfiltrate sensitive data**.

## LOG COLLECTION FOR INCIDENT RESPONSE

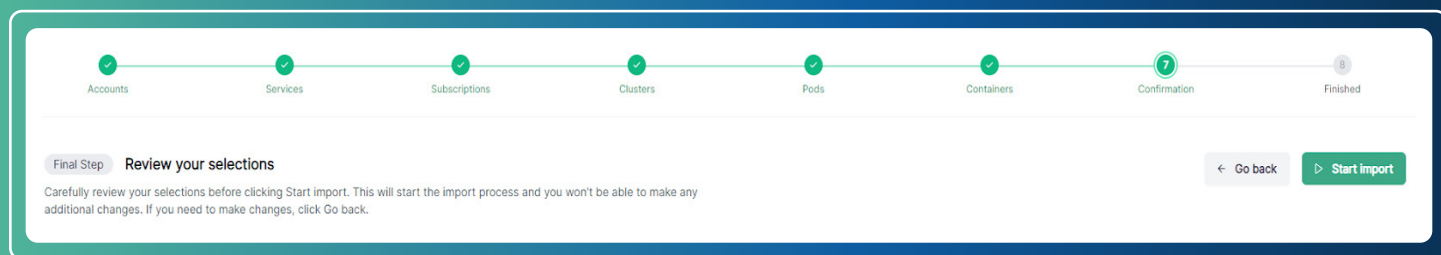
Kubernetes logs are **essential** for analyzing security events in AKS.

### Key Logs to Collect:

#### Kubernetes API Server Logs

In AKS the control plane (including the API server) is fully managed by Microsoft. Customers do not have direct access to these pods or their logs via `kubectl` because they are not deployed in customer-accessible namespaces. Instead, control plane events are available through Azure Monitor (when properly configured) or via audit logs if enabled.

- **Audit Logs** (Tracks configuration changes and user actions)  
`kubectl get --raw "/apis/audit.k8s.io/v1/events"`
- **Application Logs** (Collected via Azure Monitor & Log Analytics)
  - Enable log collection:  
`az monitor log-analytics workspace create --resource-group <RESOURCE_GROUP> --workspace-name <WORKSPACE_NAME>`
  - **Stream logs to Log Analytics:**  
`az aks enable-addons --addons monitoring --name <CLUSTER_NAME> --resource-group <RESOURCE_GROUP>`



Cado will automatically collect all key logs and forensic artifacts from an AKS container for investigation. For most acquisitions, the import and processing will take just a few minutes to complete.

For more information: [Azure Monitor for Containers](#), [Cado Azure AKS](#)

## CONTAINING A COMPROMISED AKS CLUSTER

If an attacker compromises a Kubernetes pod, immediate containment is necessary.

### Steps to Isolate a Compromised Pod:

1. **Identify the compromised pod:**

```
kubectl get pods --all-namespaces --sort-by=.status.startTime
```

2. **Delete the compromised pod:**

```
kubectl delete pod <POD_NAME> --namespace=<NAMESPACE>
```

3. **Use Network Policies to restrict external communication:**

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: deny-egress
  namespace: default
spec:
  podSelector: {}
  policyTypes:
  - Egress
  egress: []
```



For more information: [Azure Kubernetes Security Best Practices](#)



# Automating Incident Response

Incident response automation helps security teams **reduce response times, minimize manual effort, and improve accuracy** in detecting and mitigating security threats. Azure provides various automation capabilities that streamline incident response tasks, including **isolating compromised VMs, collecting forensic artifacts, and analyzing logs for Indicators of Compromise (IOCs)**.

## Automating VM Isolation

When a Virtual Machine (VM) is compromised, isolating it quickly prevents lateral movement and data exfiltration. **Azure Automation** allows security teams to trigger automated isolation workflows based on alerts from **Microsoft Defender for Cloud** or SIEM systems like **Microsoft Sentinel**.

### STEPS TO AUTOMATE VM ISOLATION:

1. **Define an Automation Runbook** to update the VM's **Network Security Group (NSG)** and deny all inbound/outbound traffic.
2. **Trigger the automation** when an alert is generated for a high-risk event.
3. **Monitor and log the isolation event** for forensic analysis.

Azure CLI Example: Restricting VM Network Access

```
az network nsg rule create --resource-group <RESOURCE_GROUP>
--nsg-name <NSG_NAME> \

--name "DenyAllTraffic" --priority 100 --direction Inbound
--access Deny \

--protocol "*" --destination-port-range "*"

```

More Information: [Azure Network Security Group Documentation](#)



# Automating Memory Dump Collection for Forensics

Memory analysis is crucial for detecting in-memory malware, credential dumping, or active attacker sessions. **Azure Automation** and **Azure Runbooks** can automate the **collection of memory dumps from suspicious VMs**.

## STEPS TO AUTOMATE MEMORY DUMP COLLECTION:

- **Use Azure Automation to execute a script** that captures a memory dump of a running VM.
- **Store the memory dump securely in Azure Blob Storage** for forensic analysis.
- **Trigger this workflow automatically** based on threat detections.

PowerShell Example: Collecting a Memory Dump from an Azure VM using DumpIT

```
Invoke-AzVMRunCommand -ResourceGroupName <RESOURCE_GROUP> -VMName  
<VM_NAME> \  
  
-CommandId 'RunPowerShellScript' -ScriptPath  
'C:\Tools\DumpIt.exe'
```

More Information: [Azure Run Command Documentation](#)



# Automating Log Analysis for Indicators of Compromise (IOCs)

Detecting **Indicators of Compromise (IOCs)** in logs is a key part of incident response. Security teams can **use Azure Monitor and Log Analytics to automate IOC searches** across Azure environments.

## STEPS TO AUTOMATE IOC DETECTION:

1. **Ingest logs from critical Azure services** (Activity Logs, Resource Logs, NSG Flow Logs, Defender for Cloud Alerts) into **Microsoft Sentinel**.
2. **Create a KQL (Kusto Query Language)** script to scan logs for known IOCs (e.g., malicious IPs, suspicious PowerShell execution).
3. **Set up automated alerts and response actions** when matches are found.

Example KQL Query: Searching for Malicious IPs in NSG Flow Logs

```
AzureDiagnostics  
  
| where Category == "NetworkSecurityGroupFlowEvent"  
  
| where RemoteIP in ("203.0.113.5", "198.51.100.8")  
  
| project TimeGenerated, RemoteIP, DestinationPort
```

More Information: [Microsoft KQL Query Guide](#)



## Using Azure Automation for Incident Response

Azure provides multiple **automation tools** that security teams can use to orchestrate incident response workflows:

Azure Automation Tool	Use Case
Azure Automation Runbooks	Automate tasks such as <b>isolating VMs, collecting artifacts, and resetting passwords.</b>
Logic Apps	Orchestrate complex workflows and integrate <b>SIEM/SOAR tools.</b>
Microsoft Sentinel Playbooks	Automate responses like <b>blocking malicious IPs, disabling compromised accounts.</b>
Azure Functions	Run custom scripts triggered by <b>security alerts.</b>

More Information: [Azure Automation Overview](#)



# Forensic Analysis in Azure



Forensic analysis is crucial for understanding security incidents, identifying root causes, and determining the extent of a breach. Azure provides several built-in tools and services to support forensic investigations, including **Network Security Group (NSG) Flow Logs**, **Azure Network Watcher**, and **Microsoft Sentinel**.

## Analyzing Network Traffic

Monitoring and analyzing network traffic is key to detecting lateral movement, command-and-control communications, and data exfiltration attempts.

### USING NETWORK SECURITY GROUP (NSG) FLOW LOGS

NSG Flow Logs record network activity for resources in a **Virtual Network (VNet)** and help security teams analyze suspicious traffic patterns.

#### Steps to Analyze NSG Flow Logs:

1. **Enable NSG Flow Logs** in Azure Network Watcher.
2. **Store logs in an Azure Storage account or send them to Log Analytics.**
3. **Analyze logs with KQL queries in Azure Monitor.**

Example KQL Query: Identifying Suspicious Traffic

```
AzureDiagnostics
| where Category == "NetworkSecurityGroupFlowEvent"
| where Action == "Deny"
| project TimeGenerated, SourceIP, DestinationIP, DestinationPort
```

More information: [NSG Flow Logs](#)

## USING AZURE NETWORK WATCHER FOR PACKET CAPTURE

Azure **Network Watcher** provides deep network inspection through packet capture.

### Steps to Capture Network Traffic:

1. **Start a packet capture session** on a specific VM:

```
az network watcher packet-capture create \  
--resource-group <RESOURCE_GROUP> --vm <VM_NAME> \  
--name PacketCaptureSession --storage-account <STORAGE_ACCOUNT>
```

2. **Download, and analyze the packet capture** using tools like [Wireshark](#)

More information: [Azure Network Watcher Packet Capture](#)



# Using Microsoft Sentinel for Forensic Analysis

Microsoft Sentinel is a **cloud-native SIEM and SOAR platform** that helps correlate and investigate security incidents across multiple data sources.

## KEY FORENSIC CAPABILITIES IN SENTINEL:

1. **Log correlation:** Collects logs from Azure AD, NSG Flow Logs, VM logs, and Microsoft Defender for Cloud.
2. **Hunting queries:** Uses **KQL queries** to search for anomalous activity.
3. **Incident enrichment:** Correlates alerts with threat intelligence feeds.

Example KQL Query: Detecting Suspicious PowerShell Activity

```
SecurityEvent
| where EventID == 4688
| where ProcessName contains "powershell.exe"
| where CommandLine contains "Invoke-WebRequest"
| project TimeGenerated, Account, ProcessName, CommandLine
```

More information: [Hunting Threats in Microsoft Sentinel](#)



# Post-Incident Review and Continuous Improvement

A well-structured **post-incident review** ensures that organizations learn from incidents and refine their response strategies. Conducting a **lessons learned session** and maintaining **detailed documentation** improves future response capabilities.

## Conducting a Lessons Learned Session

A **lessons learned session** helps security teams reflect on an incident and improve future readiness.

### STEPS TO FACILITATE A LESSONS LEARNED SESSION:

1. **Gather all relevant stakeholders**, including security, IT, legal, and business leaders.
2. **Review the timeline of events** and response actions.
3. **Identify gaps in detection, response, or containment.**
4. **Document key takeaways and update policies accordingly.**
5. **Create action items** for improvements in tools, training, or processes.

More information: [Microsoft Incident Response Best Practices](#)

## Documenting the Incident Response Process

Proper documentation ensures that future investigations are more efficient and that response teams can apply insights to new threats.

### KEY INFORMATION TO DOCUMENT:

- **Incident Summary:** Attack vector, affected resources, and timeline.
- **Detection & Response:** How the attack was detected and what actions were taken.
- **Root Cause Analysis:** Underlying vulnerabilities or misconfigurations.
- **Lessons Learned:** What worked well and what needs improvement.
- **Recommendations:** Action items for enhancing security posture.

Maintaining an **incident response knowledge base** improves team efficiency and readiness for future threats.

More information: [NIST Guide to Incident Response Documentation](#)

# Common Challenges in Azure Incident Response



Despite strong security tools, **Azure incident response** faces unique challenges that require **strategic planning and proactive mitigation**.

## Data Volatility



Azure's **dynamic cloud environment** means resources may be **created and terminated frequently**, making forensic data collection challenging.


### MITIGATION STRATEGIES:

- **Implement automated evidence collection** workflows to capture logs and snapshots immediately after detection.
- **Use Azure Storage for long-term log retention** beyond default limits.

## Multi-Account Environments

Many organizations use **multiple Azure subscriptions and accounts**, which can complicate centralized security monitoring.

### MITIGATION STRATEGIES:

- **Use Azure Management Groups** to apply consistent security policies across subscriptions.
  - **Leverage Azure Lighthouse** for centralized security visibility and incident response.
- 



## Limited Forensic Capabilities in Cloud-Native Services

Some Azure services provide **minimal forensic data**, limiting deep investigations.

### EXAMPLES:

- **Azure Functions & Logic Apps:** Stateless services with limited logging.
- **AKS Containers:** Rapidly terminated workloads, leading to loss of forensic artifacts.
- **Cosmos DB & Azure SQL:** Restricted access to internal logs.

### MITIGATION STRATEGIES:

- **Enable extended logging** in Log Analytics.
- **Capture memory snapshots and process logs** for containerized workloads.
- **Use Microsoft Defender for Cloud** for real-time threat detection.



## Cross-Region Considerations

Incident response efforts can be complicated by **regional data residency laws** and **log storage limitations**.

### MITIGATION STRATEGIES:

- **Replicate logs across multiple regions** using Azure Monitor.
- **Use global security solutions like Azure Sentinel** for centralized investigations.
- **Implement encryption and pseudonymization** to meet compliance requirements.

## Human Error and Misconfigurations

Misconfigurations and accidental exposure of sensitive resources are common security issues in Azure.

### MITIGATION STRATEGIES:

- **Use Azure Policy** to enforce security best practices automatically.
- **Monitor security posture continuously with Microsoft Defender for Cloud.**
- **Regularly conduct security assessments and automated compliance checks.**

More information: [Azure Policy Documentation](#)

## Official Azure Tools

These Microsoft-provided tools help detect, investigate, and respond to security incidents in Azure environments:

- **Azure Security Center:** Provides real-time security posture management and advanced threat protection.
- **Azure Sentinel:** A **cloud-native SIEM** for threat detection, log correlation, and automated response.
- **Azure Defender:** Protects workloads such as VMs, databases, and containers with built-in threat detection.
- **Azure AD Incident Response PowerShell Module:** Facilitates **incident analysis** by querying and investigating security events in **Azure Active Directory**.





## Community Tools

In addition to Microsoft tools, several **community-driven** forensic utilities can assist in Azure security investigations:

- **Sparrow**: Developed by **CISA**, Sparrow helps analyze Azure AD and **detect compromise indicators**.
- **Mandiant Azure AD Investigator**: Identifies **suspicious activity in Azure AD**.
- **AzureHound**: Part of the **BloodHound suite**, AzureHound maps privileges and permissions across Azure environments.
- **Hawk**: A PowerShell-based tool that **collects and analyzes Azure AD and O365 logs**.
- **CrowdStrike Reporting Tool for Azure (CRT)**: Provides insights into **Azure security configurations and risk posture**.
- **Cloud Forensic Utils**: A collection of **open-source forensic** tools for analyzing Azure environments.

## Training and Resources

To enhance **incident response capabilities**, security teams should continuously train and leverage industry-recognized resources:

- **Microsoft Documentation**: Covers **Azure security best practices, forensic methodologies, and threat detection guidance**.
- **SANS Institute**: Provides **professional training on cloud incident response, threat hunting, and forensic investigations**.
- **Microsoft Learn & Certifications**: Offers **role-based security certifications**, including **Azure Security Engineer Associate (AZ-500)**.

More information: [Microsoft Security Training](#)

## Further Reading

Microsoft Provides [playbooks for specific incident response scenarios](#) in Azure, which can help security teams follow structured response workflows:

- [Phishing Investigation](#)
- [Password Spray Investigation](#)
- [Ransomware Attack](#)
- [App Consent Grant](#)
- [Compromised or Malicious Application](#)
- [Forensic / Legal Investigation](#)

### MICROSOFT SECURITY CHECKLISTS AND BEST PRACTICES

Microsoft provides a number of **security best practice guides** for securing Azure environments:

- [Azure Security Best Practices and Patterns](#)
- [Azure Operational Security Best Practices](#)
- [Security Best Practices for Azure Solutions](#)



# Conclusion and Recommendations

As cloud environments become more **dynamic and complex**, organizations must adopt a **proactive approach** to **incident response** in Azure.



## Key Takeaways



**Preparation is Critical:** Organizations should establish **robust incident response plans** and **enable forensic logging** before an incident occurs.



**Automation Enhances Response:** Leveraging **Azure Sentinel Playbooks, Logic Apps, and SOAR tools** helps accelerate incident mitigation.



**Forensic Readiness is Essential:** Ensuring access to **NSG Flow Logs, Azure AD sign-in logs, and VM snapshots** improves investigation accuracy.



**Post-Incident Learning Strengthens Security:** Continuous **review and refinement** of security processes help adapt to evolving threats.



# Recommendations for Improving Incident Response in Azure



**Adopt a Zero Trust Security Model:** Implement least privilege access, MFA enforcement, and network segmentation.



**Enhance Logging & Monitoring:** Enable and centralize logs across Azure AD, NSG Flow Logs, and Defender for Cloud.



**Utilize Incident Response Playbooks:** Create automated workflows using Azure Sentinel Playbooks and Runbooks to improve detection and response times.



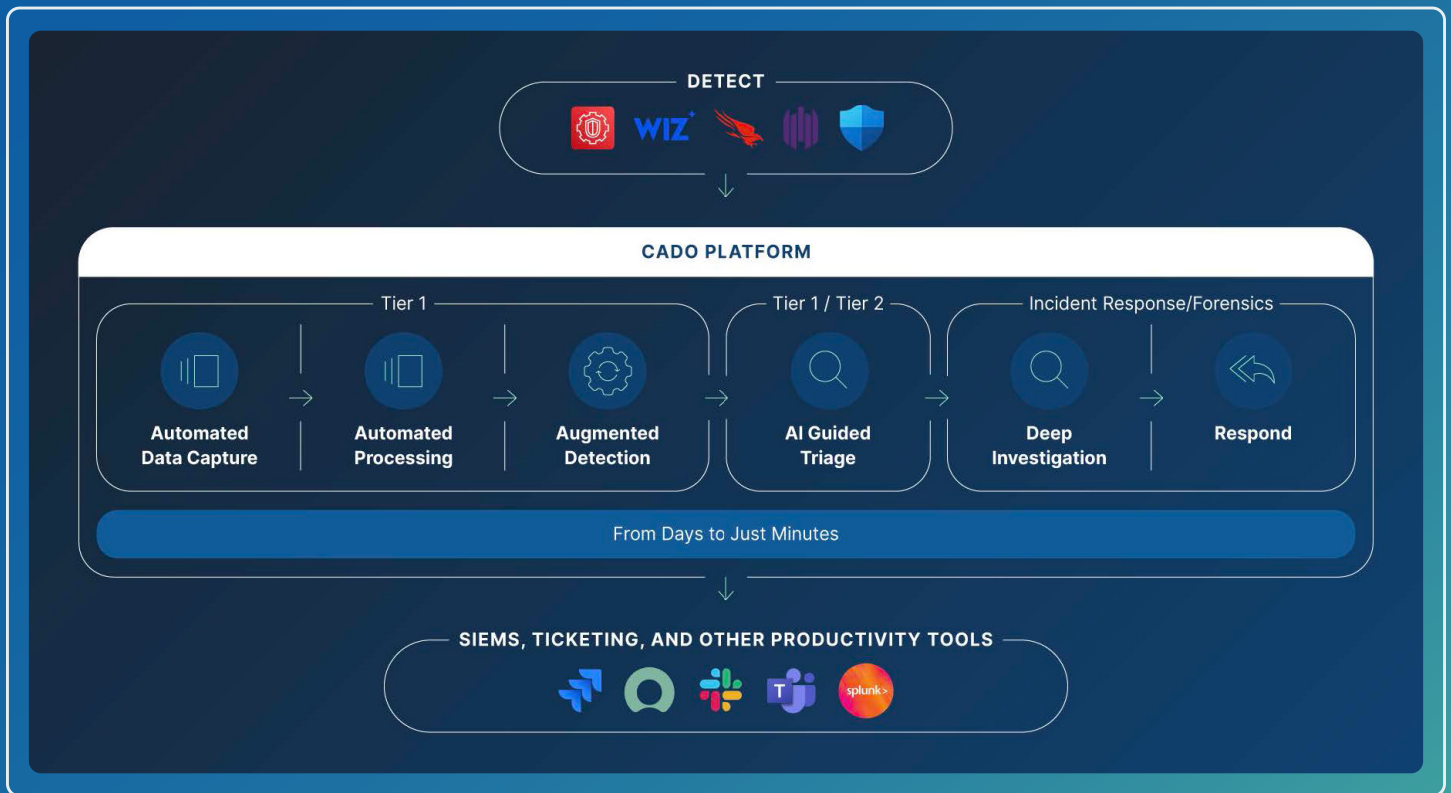
**Invest in Continuous Security Training:** Ensure security teams remain up to date with the latest Azure threat detection and forensic analysis techniques.



**Regularly Test Incident Response Plans:** Conduct tabletop exercises and live response simulations to validate security readiness.



# What is the Cado Platform?



In today's complex and evolving hybrid world, you need an investigation platform you can trust to deliver answers. Cado Security empowers teams with unrivaled data acquisition, extensive context, and unparalleled speed. The Cado Platform leverages the power of the cloud to implement a robust and repeatable investigation process.





An Overview of an Investigation in the Cado Platform

## Core Features



**Collect From Anywhere:** Achieve forensic-level visibility across your entire estate and automatically capture hundreds of data sources across cloud-provider logs, disk, memory, and more. No agent required, so zero impact to production systems.



**AI-Powered Investigations:** Leverage Cado's AI Investigator to streamline investigation and response process. Cado's local Large Language Model (LLM), AI Investigator empowers analysts to jump into a new investigation and get high-level context, fast.



**Cloud Native:** Cado is deployed natively in your cloud environment to ensure your unique data privacy requirements are met. Our patented cloud-based architecture delivers automated data collection and parallel data processing, decreasing time to investigation.



**Powerful Analytics:** Collected data is enriched using third-party and proprietary threat intelligence. Key incident details such as root cause, compromised roles and assets, and a complete timeline of events are automatically surfaced.







If you'd like to learn more about what Cado Security is doing to help advance investigations and incident response, request a demo today.

See a Demo



CADO //