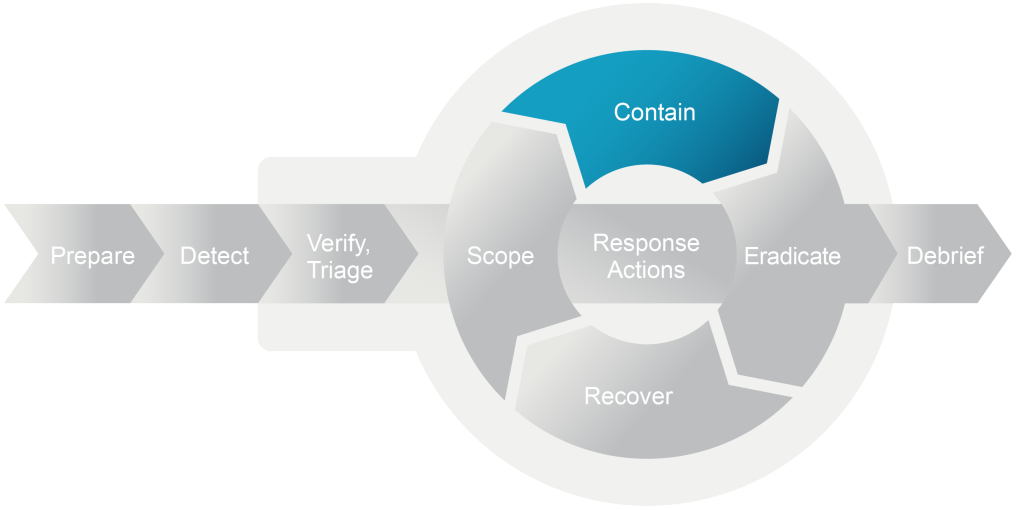
Contain: Step-by-Step



The following steps provide a condensed reference for containment activities. Each step corresponds to topics covered in the Contain Activity chapter of *Dynamic Incident Response* (www.dynamicincidentresponse.com), organized for use when stopping attacker activity, preserving evidence, and preventing further harm.

Step 1. Assess Containment Urgency and Strategy

1. Evaluate immediate containment triggers that demand urgent action, including:
   - Active data destruction or encryption in progress.
   - Ongoing data exfiltration of sensitive information.
   - Threats to critical operations or safety systems.
   - Regulatory compliance timeframes that require a rapid response.
2. Determine containment approach based on organizational context:
   - Passive containment: Monitor attacker activities to gather intelligence while limiting damage.
   - Active containment: Take decisive action to stop attacker activities, accepting the possibility of alerting the attacker.
   - Adaptive containment: Combine approaches with progressively restrictive measures based on attacker behavior.
   - Deceptive containment: Deploy honeypots or decoy systems that appear identical to production to divert the attacker while gathering intelligence and protecting real assets.
3. Balance competing priorities, including:
   - Evidence preservation needs versus operational continuity requirements.
   - Intelligence-gathering value versus the risk of continued attacker access.
   - Business impact of containment measures versus security benefits.
4. Plan coordination strategy for complex incidents, including:
   - Identify all known compromised systems, accounts, and services.
   - Coordinate timing across network, endpoint, identity, and application teams.
   - Prepare for simultaneous containment actions to prevent attacker pivoting (sequential isolation alerts adversaries and gives them time to escalate).

- Plan credential reset sequencing to prioritize the most privileged accounts first, then service accounts, then standard user accounts.
- Prepare rollback procedures for infrastructure-wide changes such as firewall rule updates or network topology modifications.
- Establish communication channels for coordinated execution.

Step 2. Implement Identity and Access Containment

1. Invalidate compromised credentials across all authentication systems for the known-compromised set (eradicate Step 4 expands credential remediation to the blast-radius accounts revealed during scope and to deeper identity primitives like KRBTGT and trust passwords), including:
   - Reset passwords for affected user accounts through the primary identity provider.
   - Rotate service account credentials, API keys, and programmatic access tokens.
   - Coordinate credential resets for accounts synced across multiple systems (on-premises and cloud).
   - Prioritize privileged accounts and high-risk compromises first.
2. Terminate active sessions to prevent continued access, including:
   - Revoke all active sessions and authentication tokens through the identity provider.
   - Force disconnect VPN connections through the concentrator management interface.
   - Terminate active RDP sessions on Windows servers using built-in session-management commands.
   - Kill active SSH sessions on Linux systems using process-management commands.
   - Invalidate browser sessions by incrementing the session token version, where the IdP supports it.
   - Contact the SaaS application support team for a forced logout when necessary.
3. Revoke refresh tokens and persistent credentials, including:
   - Invalidate OAuth refresh tokens that could generate new access tokens.
   - Revoke offline access tokens that enable access without user interaction.
   - Implement token deny lists for stateless JWT tokens where applicable.
   - Monitor authentication logs for token endpoint requests indicating cached credential use.
4. Implement conditional access policies for ongoing protection, including:
   - Deploy location-based restrictions blocking authentication from attacker geographic regions.
   - Require device compliance for authentication only from managed endpoints.
   - Enable risk-based policies that block sign-ins from suspicious IP addresses or from users exhibiting anomalous behavior.
   - Escalate MFA requirements to phishing-resistant methods.
5. Coordinate SSO and multi-application containment, including:
   - Disable compromised accounts at the IdP level to block authentication to all SSO-connected applications.
   - Verify which applications use the IdP for authentication, versus those that use local accounts, requiring separate revocation.
   - Monitor for authentication synchronization delays between the IdP and connected applications.
   - Enforce MFA requirements at the IdP level, escalating to phishing-resistant methods during containment.
6. Disable third-party integrations and automation, including:
   - Revoke OAuth-connected applications authorized by compromised accounts.
   - Disable automation rules, webhooks, and CI/CD deployment keys.
   - Invalidate API tokens authenticating programmatic access.

- Review and disable browser extensions with broad permissions.

Step 3. Implement Network and Host Containment

1. Deploy network-level isolation, including:
   - Move compromised systems to quarantine VLANs with restricted access.
   - Implement firewall rules blocking specific protocols, ports, or destinations.
   - Configure DNS sinkholes redirecting attacker domains to internal monitoring systems.
   - Apply micro-segmentation using host-based firewalls or software-defined networking.
   - Consider route manipulation for enterprise-wide containment of widespread compromises.
2. Isolate individual hosts while maintaining investigative access, including:
   - Enable EDR isolation features to quarantine systems while preserving forensic connection.
   - Configure local firewall rules to block inbound and outbound connections, except for management protocols.
   - Remove compromised instances from production scaling groups and load balancers (cloud environments).
   - Implement process termination for identified malicious processes (with caution for watchdog processes).
   - Apply application control technologies (AppLocker, Gatekeeper, AppArmor) to block execution of unauthorized executables, scripts, and libraries.
3. Apply application-level restrictions, including:
   - Deploy web application firewall rules blocking malicious request patterns.
   - Implement database access restrictions and query monitoring for unusual patterns.
   - Disable AI agent tool access and MCP server connections for compromised accounts or exploited integrations.
   - Revoke service principals or API keys granting AI systems access to organizational data (these are often separate from user account credentials).
   - Configure email flow rules to quarantine messages from compromised accounts.
   - Apply service disruption where other containment methods prove insufficient: stop web services being used for exfiltration, disable remote access protocols such as RDP or SSH that provide attacker entry points, or shut down specific compromised business applications. Coordinate service disruption with business stakeholders.

Step 4. Implement Cloud-Specific Containment

1. Verify cloud security demarcation for affected services, including:
   - Determine which containment actions the organization can implement directly and which require cloud provider engagement.
   - Do not rely on broad IaaS, PaaS, or SaaS labels to determine responsibility; verify the security demarcation for each specific product.
   - Engage cloud provider security teams for incidents affecting infrastructure outside customer control (e.g., hypervisor, physical networks).
2. Prioritize cloud identity containment, including:
   - Disable compromised IAM user access keys to prevent API calls and console access.
   - Rotate service principal credentials used by automated systems.
   - Revoke OAuth tokens and refresh tokens for cloud application access.

    - Document operational impact on downstream automation and coordinate credential updates.
3. Apply cloud network isolation controls, including:
    - Modify security group rules or network ACLs to restrict traffic.
    - Move instances to a pre-configured isolation VPC dedicated to incident response.
    - Remove instances from production load balancers while maintaining running state.
4. Enable cloud resource protection and tracking, including:
    - Apply resource tags marking compromised assets (`Status:UnderInvestigation`, `IR-Case:NNN`).
    - Enable termination protection on contained instances to prevent accidental deletion.
    - Apply deletion locks on critical resources containing evidence.
    - Create snapshots of compromised volumes and virtual machine states for forensic analysis.
5. Address SaaS platform containment limitations, including:
    - Reset passwords or set temporary random passwords, blocking account access.
    - Use platform-specific freeze features to preserve data while blocking sign-in.
    - Revoke active sessions through administrative interfaces where available.
    - Contact the SaaS provider support team for capabilities beyond the exposed administrative features.
6. Contain serverless and ephemeral resources, including:
    - Disable event triggers invoking compromised serverless functions or containers.
    - Replace function code with deny-and-log implementations that capture invocation attempts.
    - Revoke IAM roles and service permissions for serverless resources.

Step 5. Address Remote Work and Modern Environment Challenges

1. Implement endpoint-focused containment for remote workers, including:
    - Use EDR isolation features rather than network-based controls for home network devices.
    - Apply per-user or per-device VPN access policies to restrict compromised sessions without disrupting all remote workers.
    - Leverage identity-based controls and conditional access policies for SaaS applications accessed directly over the internet.
    - Coordinate with employees for secure device recovery through shipping or on-site visits.
    - Balance forensic evidence preservation against business continuity and logistical complexity.
2. Manage BYOD and personal device scenarios, including:
    - Use MDM or MAM solutions to selectively manage corporate data without affecting personal information.
    - Apply conditional access policies requiring device compliance for corporate resource access.
    - Respect privacy boundaries while maintaining organizational security requirements.
3. Account for encrypted communications challenges, including:
    - Implement TLS inspection proxies where possible to analyze encrypted communications from attackers.
    - Restrict egress TLS traffic to pass through inspection points and investigate connections that bypass them.
    - Address DNS over HTTPS (DoH) limitations by using host-based DNS overrides, local DoH servers, or browser policy controls to redirect attacker domains.
    - Balance the benefits of security inspection against privacy concerns and application compatibility issues.

4. Address non-traditional compromise scenarios, including:
    - Implement email flow redirection and conditional access for Business Email Compromise incidents.
    - Coordinate with external parties for supply chain and partner ecosystem breaches.
    - Focus containment on access control boundaries when direct system control is unavailable.
    - Revoke API access tokens to prevent further data synchronization with compromised third parties.

Step 6. Collect and Preserve Evidence

This step preserves volatile evidence; eradicate Step 1 interprets the captured artifacts during short-form investigation.

1. Determine evidence collection timing based on organizational priorities, including:
    - Weigh the loss of possible evidence due to containment actions against the risk of allowing attackers to maintain access while evidence is collected.
    - Organizations with robust logging and external data sources (network flow logs, host telemetry agents, NDR) can isolate first, then collect, reducing attacker interference.
    - Organizations without alternate data sources for volatile network connections should collect before isolation so evidence of active connections is not lost.
2. Prioritize volatile data collection immediately, including:
    - Capture memory dumps from affected systems using a whole-system memory acquisition tool appropriate to the OS.
    - Record active network connections before isolation terminates them.
    - Document running processes and services with parent-child relationships.
    - Preserve temporary files and cache data before normal system operations clear them.
3. Collect system artifacts with longer preservation timeframes, including:
    - Export registry hives identifying persistence mechanisms.
    - Preserve event logs, audit trails, and authentication records.
    - Capture file system metadata, Prefetch files, ShimCache, and AmCache records.
    - Document browser history and cache, revealing attacker reconnaissance.
4. Capture network and application evidence, including:
    - Collect packet captures from critical time periods (if storage permits).
    - Export NetFlow data, VPC Flow Logs, and firewall logs showing connection patterns.
    - Preserve DNS query logs revealing command-and-control infrastructure.
    - Collect web server logs, database transaction logs, and application audit trails.
    - Export cloud service audit logs before retention policies delete them.

Step 7. Validate Containment Effectiveness

1. Monitor network communications for continued attacker activity, including:
    - Review firewall logs, proxy logs, and DNS queries for connections to attacker infrastructure.
    - Watch for new communication patterns indicating alternative command-and-control channels.
    - Validate containment using external data sources (network flow logs, host telemetry agents, NDR tools) when network isolation severs direct connections to contained systems.
    - Combine network monitoring from appliances with live investigation of contained systems.
    - Compare pre-containment and post-containment traffic patterns.

2. Verify malicious processes have ceased, including:
    - Monitor process creation on contained systems using EDR or host telemetry data.
    - Track unusual parent-child process relationships and uncommon executable paths.
    - Watch for new service installations representing attacker persistence.
    - Use process inspection utilities for baseline comparison.
3. Analyze logs for indicators of persistent compromise, including:
    - Monitor authentication logs for failed attempts to gain access using alternative credentials.
    - Watch for privilege escalation attempts that suggest local, persistent access.
    - Identify unusual file access patterns indicating continued data exfiltration.
    - Review both the system logs and external logging sources.
4. Establish baselines and compare pre-containment and post-containment activity, including:
    - Document expected process activity on contained systems and investigate deviations.
    - Compare network traffic, process execution, and log activity before and after containment.
    - Verify that malicious activities have ceased rather than shifted to different techniques or alternative infrastructure.

Step 8. Document Containment Actions and Communicate Status

1. Document containment decisions with complete context, including:
    - Record rationale for containment strategy selection (passive, active, or adaptive).
    - Timestamp each containment measure deployment with the responsible personnel.
    - Document which systems, accounts, and services were affected by each action.
    - Capture the options considered and the reasoning behind the chosen approach.
    - Maintain the chain of custody for all collected evidence.
2. Assess and document operational impact, including:
    - Identify business functions affected by each containment action.
    - Estimate user count experiencing service disruptions.
    - Calculate revenue impact from system downtime where applicable.
    - Document workarounds implemented to maintain critical operations.
    - Gather feedback from business unit leaders for future planning.
3. Communicate appropriately with diverse stakeholders, including:
    - Provide executive summaries focusing on business impact, risk reduction, and resource needs.
    - Deliver technical briefings with implementation details for IT and security teams.
    - Notify affected users about service disruptions, workarounds, and resolution timelines.
    - Coordinate with legal, compliance, and public relations teams on regulatory obligations.
4. Prepare for subsequent incident response phases, including:
    - Organize collected evidence for eradication planning and threat intelligence analysis.
    - Create a system inventory prioritizing the recovery sequence.
    - Surface containment-phase feedback (control gaps, playbook issues, tooling limitations) to debrief Step 8 rather than acting on them within the contain activity.
    - Identify additional scoping needs revealed during containment activities.