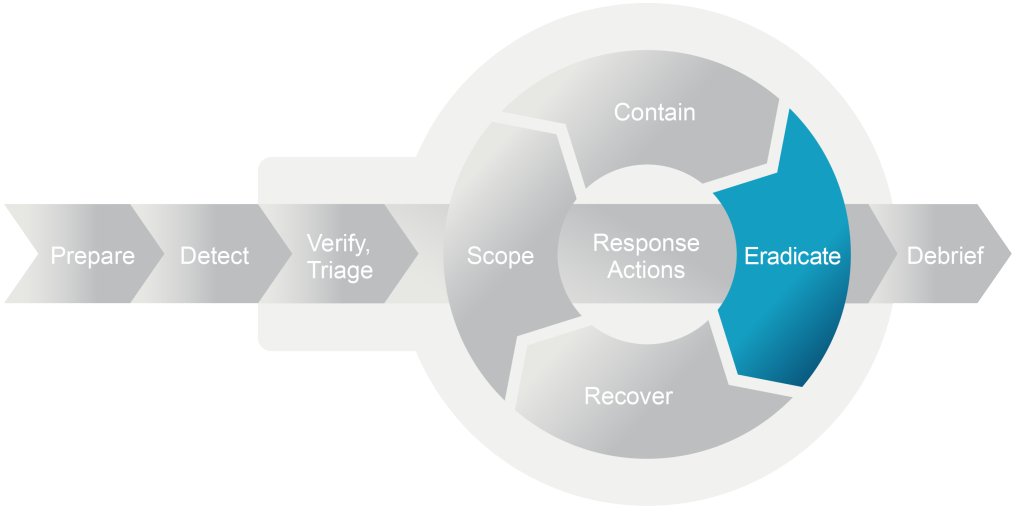
# Eradicate: Step-by-Step




The following steps provide a condensed reference for eradication activities. Each step corresponds to topics covered in the Eradicate Activity chapter of *Dynamic Incident Response* (www.dynamicincidentresponse.com), organized for use when investigating the presence of attackers, removing persistence mechanisms, and remediating the conditions that enabled the attack.

## Step 1. Conduct Short-Form Investigation to Inform Eradication

Short-form investigation provides the insight needed to proceed with eradication without waiting for the full forensic picture. A parallel long-form investigation may continue throughout eradication and into recovery to support regulatory reporting, legal proceedings, and organizational learning; the eradicate chapter's "Long-Form Investigation for Comprehensive Understanding" section describes the parallel track in detail.

1. Answer important eradication questions through evidence analysis, including:
   - What was the initial access vector? Has it been closed?
   - What credentials were compromised? Have they been rotated?
   - What other systems did the attacker access? Are they also targeted for eradication?
   - What persistence mechanisms did the attacker deploy? Are they all identified?
   - What vulnerability enabled the attack? Has it been patched?
2. Apply log investigation techniques, including:
   - Use a portable detection-rule format with a fast event-log scanner for rapid Windows Event Log analysis.
   - Query SIEM platforms for indicators of compromise across all ingested log sources.
   - Correlate authentication logs, application logs, and network flow data.
   - Add eradication-sequencing markers to the attack timeline produced during scope, identifying when each persistence mechanism was deployed so removal can proceed in safe order. Building the canonical attack-progression timeline is a scope activity; eradicate annotates it.
3. Conduct live investigation on contained systems, including:
   - Enumerate running processes, network connections, and system configuration.
   - Apply differential analysis comparing the current state against known-good baselines to identify

new services, scheduled tasks, and accounts.
    - Document findings systematically for eradication planning.
4. Perform a memory investigation when deeper analysis is required, including:
    - Analyze the memory captured during contain using a memory forensics framework to examine processes, network connections, and loaded drivers. Volatile data collection is itself a contain activity. If memory was not captured during contain, capture it now with a whole-system memory acquisition tool before proceeding with analysis.
    - Identify injected code using process-memory analysis techniques.
    - Extract command-line arguments and handles to understand attacker objectives.
5. Analyze network data to map attacker communications, including:
    - Review packet captures, NetFlow data, and firewall logs for lateral movement patterns.
    - Examine DNS logs for command-and-control domain lookups.
    - Visualize cloud network connection graphs from flow log data.
    - Identify data exfiltration indicators through volume analysis and destination review.
6. Use EDR platforms for centralized endpoint investigation, including:
    - Query EDR consoles to scope attacker activity across multiple endpoints.
    - Correlate EDR alerts with findings from log, memory, and network investigation.
7. Conduct malware investigation when malicious files are identified, including:
    - Perform static analysis: file hashing, string extraction, PE structure analysis, and cross-referencing with threat intelligence platforms.
    - Perform dynamic analysis in isolated environments: execute malware with system-monitoring tools and review sandbox reports from automated analysis platforms.
    - Identify artifacts created by the malware (files, registry keys, processes) and persistence mechanisms requiring removal.
    - Extract network IOCs (C2 domains, IP addresses) for containment and continued scoping.
8. Investigate Business Email Compromise (BEC) when email-based attacks are suspected, including:
    - Enumerate mailbox rules and forwarding configurations on affected accounts.
    - Review organization-level mail flow and transport rules for unauthorized changes.
    - Enumerate OAuth application permissions and third-party integrations.
    - Analyze sign-in logs for impossible travel patterns and suspicious authentication activity.
    - Coordinate with finance and accounting teams to trace fraudulent transactions during the compromise window.
9. Investigate insider threats when authorized users are involved, including:
    - Coordinate closely with HR, legal, and company leadership before beginning the investigation.
    - Focus on data access patterns, privilege changes, and exfiltration methods.
    - Look for remote access tools, secondary accounts, or modified system configurations that could allow continued access.
    - Document findings meticulously for potential legal proceedings.
10. Investigate supply chain compromises when trusted channels are involved, including:
    - For software supply chain incidents, identify all systems with the compromised software version and review vendor IOCs.
    - For hardware supply chain incidents, document serial numbers, procurement sources, and shipping chains.
    - For service provider compromises, identify all access points the compromised partner has to the environment and review authentication logs for unusual activity.
    - Coordinate with the compromised supplier or partner to share investigation findings.

11. Investigate cloud environments for IaaS and SaaS compromises, including:
    - Review cloud audit logs for unauthorized API calls and IAM modifications.
    - Apply differential analysis comparing the deployed state against IaC definitions or documented baselines.
    - Investigate IAM roles, assume-role trust relationships, and cross-account access for unauthorized changes.
    - Review storage access logs, serverless function deployments, and container image modifications.
    - For SaaS compromises, request activity reports from the provider and review integration logs from connected on-premises systems.
    - Enumerate privileges and OAuth applications within SaaS environments for persistence mechanisms.

## Step 2. Perform Root Cause Analysis

1. Identify the root cause of the incident, including:
    - Trace the attack chain back to the initial compromise point.
    - Distinguish between immediate causes and underlying root causes.
    - Document the sequence of events that enabled the attacker's access.
    - Identify systemic weaknesses that allowed the attack to succeed.
2. Use structured analysis techniques, including:
    - Apply fishbone diagram mapping to categorize contributing factors across the four P's: People, Process, Product, and Policy.
    - Alternatively, use the Five Whys technique for focused, linear root cause analysis.
    - Engage relevant stakeholders from IT, security, and business units.
    - Validate findings against collected evidence.
3. Document root cause findings for remediation planning, including:
    - Record specific vulnerabilities, misconfigurations, or process failures.
    - Identify preventive measures to address each contributing factor.
    - Prioritize remediation based on risk and feasibility.
    - Feed findings into both immediate eradication and long-term security improvements.

## Step 3. Remove Persistence Mechanisms

1. Address Windows persistence mechanisms, including:
    - Use a consolidated autostart enumeration tool to list persistence locations in a single view and compare against known-good baselines.
    - Examine registry Run keys (HKLM and HKCU `\Software\Microsoft\Windows\CurrentVersion\Run`).
    - Review scheduled tasks for unauthorized entries.
    - Check services for unauthorized entries.
    - Examine startup folders, WMI event subscriptions, and DLL search order hijacking.
    - Review the Group Policy for malicious scripts or software deployment.
2. Address Linux persistence mechanisms, including:
    - Review cron jobs in `/etc/crontab`, `/etc/cron.d/`, and user crontabs, and review pending at jobs.
    - Examine systemd services in `/etc/systemd/system/` and user service directories.
    - Check shell initialization files (`.bashrc`, `.profile`, `/etc/profile.d/`).

    - Review `authorized_keys` files for unauthorized SSH access.
    - Examine kernel module and library preloading configurations (`/etc/ld.so.preload` and `LD_PRELOAD`).
    - Review PAM modules in `/etc/pam.d/` for backdoors.
    - Check for unauthorized SUID/SGID binaries and sudo configuration changes.
    - Review package manager hooks, git hooks, and udev rules for malicious entries.
3. Remove service and application persistence mechanisms, including:
    - Search for files with suspicious characteristics (encoded content, eval functions).
    - Compare web directories against known-good baselines.
    - Review web server logs for access patterns to suspicious files.
    - Verify file integrity against deployment manifests or version control.
4. Address cloud persistence mechanisms, including:
    - Review IAM users, roles, and policies for unauthorized access grants.
    - Examine serverless functions, container images, and event triggers.
    - Check for unauthorized API keys, access tokens, and service account credentials.
    - Review resource policies, bucket policies, and cross-account access configurations.
    - Audit OAuth application registrations and consent grants.

## Step 4. Remediate Accounts and Identity Systems

This step addresses the identity system as a whole, including the blast-radius accounts revealed during scope and the deeper identity primitives (KRBTGT, trust passwords, delegation, hybrid AD/Entra) that are out of scope for contain. The known-compromised set is already invalidated during contain; Step 4 picks up from there and works outward.

1. Remediate local accounts and credentials across the expanded scope identified after contain, including:
    - Remove unauthorized local accounts from affected systems.
    - Reset passwords for additional legitimate accounts revealed during scope as having been touched, beyond the known-compromised set already reset during contain.
    - Clear cached credentials from LSASS, browser stores, and credential managers.
    - Rotate local administrator passwords using a managed local-administrator password solution.
2. Remediate Active Directory accounts and credentials, expanding remediation across the AD environment, including:
    - Follow Microsoft's tier model (Tier 0, Tier 1, Tier 2) for prioritization.
    - Reset passwords for compromised accounts, starting with those with the highest privilege, including accounts revealed during scope that contain did not address.
    - Review and remove unauthorized group memberships.
    - Audit and reset service account credentials with application team coordination.
3. Reset the KRBTGT account when a Kerberos compromise is suspected:
    - Perform the KRBTGT password reset twice, with ten-plus-hour delay between resets.
    - Verify replication completion across all domain controllers after each reset.
    - In multi-domain forests, reset the KRBTGT in the child domains before resetting it in the parent domains.
    - Monitor for authentication failures indicating active Golden Ticket usage.
    - Document reset timing for compliance and incident records.
4. Reset domain controller machine account passwords:
    -

Reset each domain controller's machine account password individually after completing KRBTGT resets.
    - Allow replication to complete between machine account resets.
5. Reset trust passwords when inter-domain or inter-forest trusts are involved:
    - Reset trust passwords on the trusting side of each affected trust relationship.
    - Verify trust functionality after reset.
6. Disable unconstrained delegation where not strictly required, including:
    - Identify computer and user objects with unconstrained delegation using directory-query tooling appropriate to the environment.
    - Disable unconstrained delegation on objects that do not require it.
    - Migrate to constrained or resource-based constrained delegation where delegation is needed.
7. Address hybrid and cloud identity systems, including:
    - Coordinate remediation across on-premises AD and Entra ID.
    - Revoke all refresh tokens and active sessions for compromised users in the cloud IdP.
    - Reset passwords in both environments, accounting for synchronization delays.
    - Review and remove malicious Entra ID application registrations.
    - Verify that the Entra Connect synchronization rules have not been modified and reset the synchronization service account credentials.
    - Audit federated identity provider configurations for unauthorized changes.
8. Extend session and token revocation to the expanded account set (contain Step 2 already revoked sessions and tokens for the known-compromised set), including:
    - Force termination of active sessions through the identity provider for additional accounts identified during scope.
    - Revoke OAuth refresh and access tokens, personal access tokens, and API keys for the expanded account set.
    - Rotate service credentials and API keys for service accounts revealed during scope, and update dependent applications.
    - Clear browser-stored credentials, authentication cookies, and session tokens on systems newly identified as touched by the attacker.
    - Monitor for token reuse attempts across both the contain-era and eradicate-era revocation sets.

## Step 5. Execute Targeted Removal or Rebuild

1. Choose an appropriate eradication strategy:
    - Targeted removal: Remove specific malware and persistence when the scope is well-understood.
    - Full rebuild: Reinstall from clean media when the compromise scope is uncertain.
    - Restore from backup: Use verified, clean backups when available and validated.
2. Validate the eradication method, including:
    - Confirm a clean backup exists for the backup-restore path (the per-system backup integrity validation at restore time is a recover Step 1 activity).
    - Ensure root cause is addressed before rebuilding or restoring systems to prevent reinfection.
    - Test rebuilt systems in an isolated environment before handing them to recover for production validation testing.
    - Document the eradication method and validation steps for each system.

## Step 6. Remediate Vulnerabilities

1. Identify and patch exploited vulnerabilities, including:
    - Map exploited vulnerabilities to CVE identifiers where applicable.
    - Cross-reference with the CISA Known Exploited Vulnerabilities catalog.
    - Prioritize patches for vulnerabilities actively exploited in the incident.
    - Test patches in a non-production environment before broad deployment.
2. Implement patch management during eradication, including:
    - Track patch status with inventory management tools.
    - Balance change management procedures with the urgency of eradication.
    - Consider phased rollout for large environments.
    - Verify system functionality after patching.
3. Address unpatchable systems, including:
    - Implement compensating controls (network segmentation, enhanced monitoring).
    - Document compensating controls as temporary measures requiring review.
    - Establish a timeline for system replacement or upgrade.
    - Include unpatchable systems in ongoing vulnerability management tracking.
4. Conduct broader vulnerability assessment, including:
    - Scan the environment for related vulnerabilities beyond the immediate incident scope using vulnerability assessment tooling.
    - Review configuration hardening against CIS Benchmarks or similar standards.
    - Enumerate accessible services using port scanning or local service enumeration and disable unnecessary services.
    - Review network device configurations for overly permissive firewall rules, obsolete entries, and unnecessary exposure.
    - Disable unnecessary remote access services such as Telnet, FTP, RDP, and SMB unless explicitly required.

## Step 7. Address Eradication Challenges

1. Investigate living-off-the-land techniques, including:
    - Review use of legitimate built-in system utilities commonly abused for living-off-the-land techniques.
    - Analyze command-line arguments for administrative tools.
    - Establish baselines for normal administrative tool usage.
    - Implement enhanced logging for commonly abused utilities.
2. Address fileless malware, including:
    - Focus memory analysis on identifying injected code and reflective loading.

    - Review script execution logs (PowerShell Script Block Logging, WMI traces).
    - Examine registry-resident malware and WMI persistence.
    - Clear memory-resident threats through controlled system restarts.
3. Investigate legitimate remote access tool abuse, including:
    - Audit installed remote monitoring and management (RMM) tools.
    - Identify unauthorized installations of remote access utilities commonly leveraged by attackers.
    - Review authorized tool configurations for unauthorized access grants.
    - Implement allowlisting to prevent the unauthorized installation of remote access tools.
4. Address cross-trust boundary persistence, including:
    - Enumerate forest trusts, external trusts, and federated identity relationships.
    - Review authentication logs in trusted environments for suspicious cross-boundary activity.
    - Coordinate eradication with administrators of trusted domains and identity providers.
    - Verify removal of cross-boundary artifacts in all affected environments.

## Step 8. Validate Eradication Success

1. Verify persistence mechanism removal, including:
    - Re-scan systems for indicators of compromise identified during the investigation.
    - Confirm scheduled tasks, services, and registry entries are removed.
    - Validate account remediation by monitoring authentication logs.
    - Test that blocked network indicators generate alerts if accessed.
2. Monitor for signs of continued attacker activity, including:
    - Watch for authentication attempts using revoked credentials.
    - Monitor network traffic for connections to known attacker infrastructure.
    - Review process creation logs for suspicious execution patterns.
    - Implement canary files or honeypot credentials to detect residual access.
3. Confirm vulnerability remediation, including:
    - Verify patches are successfully installed on affected systems.
    - Test compensating controls for unpatchable systems.
    - Validate that configuration hardening changes are in effect.
    - Confirm the initial access vector is closed.

## Step 9. Document Eradication Actions

1. Record investigation findings, including:
    - Document identified indicators of compromise and persistence mechanisms.
    - Record root cause analysis results and contributing factors.
    - Capture timeline of attacker activity reconstructed from evidence.
    - Note any gaps in evidence or areas requiring further investigation.
2. Document remediation actions, including:
    - Record each persistence mechanism removed with a timestamp and method.
    - Document credential resets, including accounts, timing, and coordination.
    - Capture system restoration decisions and validation results.
    - Record vulnerability patches applied and compensating controls implemented.

3. Communicate eradication status to stakeholders, including:
    - Provide an executive summary of eradication activities and outcomes.
    - Deliver technical briefings to IT teams responsible for ongoing monitoring.
    - Coordinate with legal and compliance on documentation requirements.
    - Prepare handoff documentation for recover activity follow-on work.
4. Capture eradicate-phase feedback for the debrief consolidation, including:
    - Document detection gaps that allowed initial compromise.
    - Note investigation challenges and tool limitations encountered during eradication.
    - Surface security control gaps and playbook gaps revealed during eradication; debrief Step 8 converts these into prioritized organizational improvements rather than each phase implementing them independently.