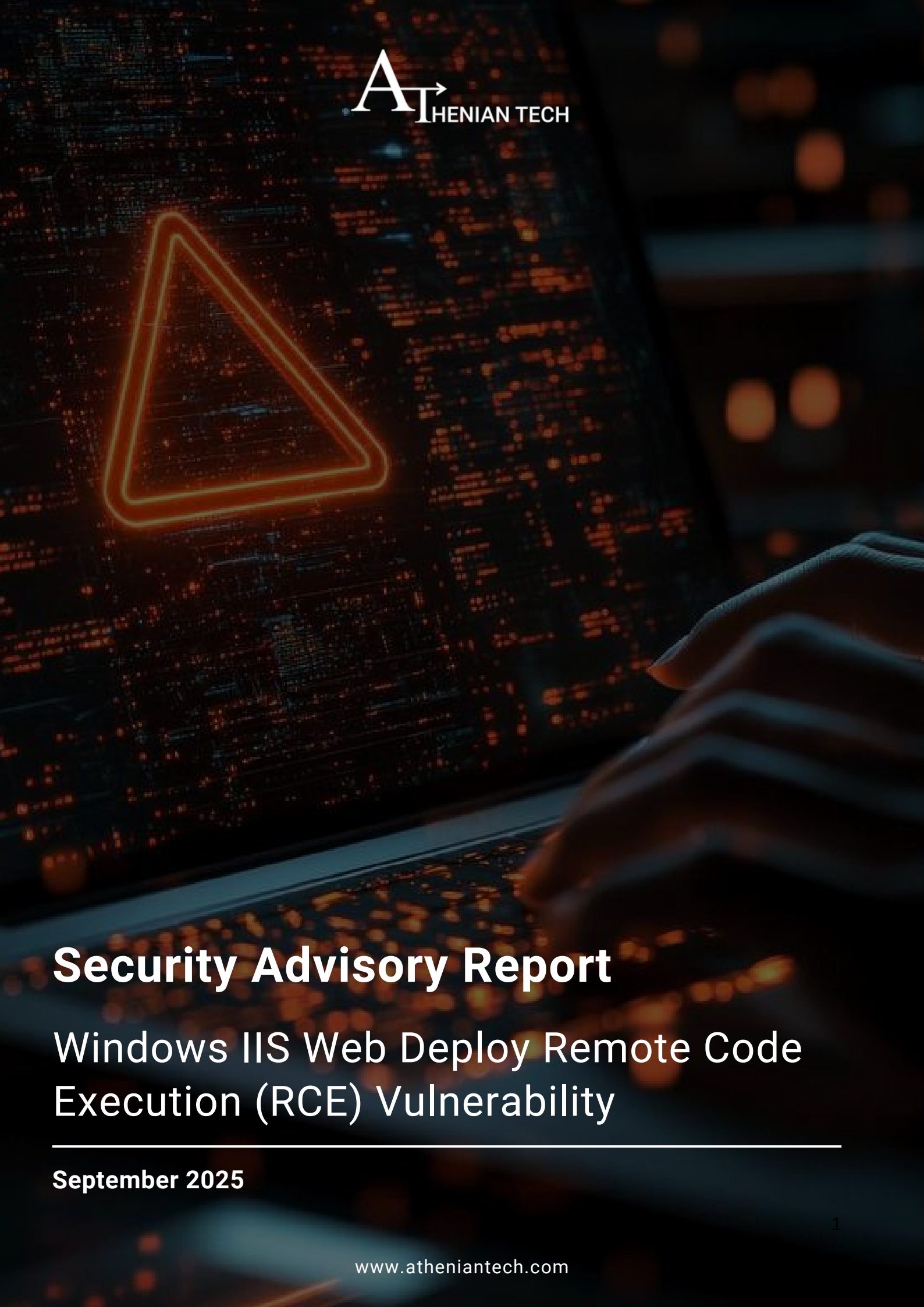

Security Advisory Report

Windows IIS Web Deploy Remote Code Execution (RCE) Vulnerability

September 2025

Table of Contents

Executive Summary

A critical remote code execution (RCE) vulnerability, tracked as CVE-2025-53772, has been identified in Microsoft's Web Deploy 4.0 tool. The flaw carries a CVSS score of 8.8 (High) and arises from the insecure deserialization of untrusted data within the msdeployagentservice and /msdeploy.axd endpoints.

By exploiting this weakness, an authenticated attacker with low privileges can craft and transmit a malicious payload in the MSDeploy.SyncOptions HTTP header. Once received, the server processes the payload by decoding, decompressing, and deserializing it using .NET's BinaryFormatter, which lacks type whitelisting. This permits execution of arbitrary system commands on the targeted IIS server, effectively granting the attacker remote control.

The disclosure of a public proof-of-concept (PoC) exploit has escalated the risk profile of this vulnerability, lowering the barrier for exploitation and increasing the likelihood of widespread attacks. Successful exploitation could enable adversaries to compromise deployment servers, tamper with web applications, exfiltrate sensitive data, and disrupt critical business services.

Microsoft has released a security update (version 10.0.2001) for Web Deploy 4.0 to remediate the issue and strongly advises immediate patching. In addition to applying updates, organizations are urged to enforce strict access controls on Web Deploy endpoints, disable unused services such as MsDepSvc, and monitor IIS logs for suspicious SyncOptions header activity.

Given Web Deploy's extensive use in DevOps and CI/CD pipelines, this vulnerability poses a systemic risk that extends beyond individual servers to entire deployment workflows. The presence of a working exploit underscores the urgency for administrators to implement patches and adopt defense-in-depth strategies without delay.

Vulnerability Details

The vulnerability, tracked as CVE-2025-53772, affects Microsoft's Web Deploy 4.0 tool, which is commonly used by administrators and DevOps teams to synchronize, package, and deploy web applications and IIS configurations. The flaw is rooted in the insecure deserialization of untrusted data, a long-standing security weakness categorized under CWE-502 (Deserialization of Untrusted Data).

CVE Information

- **CVE ID:** CVE-2025-53772
- **CWE Classification:** CWE-502 – Deserialization of Untrusted Data
- **CVSS v3.1 Base Score:** 8.8 (High)
- **CVSS Vector:** AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
- **Severity:** High
- **Exploitability:** Network-based, low attack complexity, no user interaction required, low-privilege authenticated access required
- **Disclosure Date:** August 12, 2025
- **Patch Released:** Security update version 10.0.2001 for Web Deploy 4.0

Affected Components

- **Endpoints:**
  - **/msdeploy.axd** (exposed via Web Management Service, WMSvc)
  - **msdeployagentservice** (Web Deploy Agent Service, MsDepSvc)
- **Product:** Microsoft Web Deploy 4.0
- **Fixed Version:** 10.0.2001 or later

Root Cause

The vulnerability lies in the processing of the MSDeploy.SyncOptions HTTP header. When a client communicates with Web Deploy, this header is expected to contain deployment configuration parameters in a serialized format.

The vulnerable code path performs the following operations:

1. **Base64 Decoding** – The header value is decoded from Base64.
2. **GZip Decompression** – The decoded content is decompressed.
3. **.NET BinaryFormatter Deserialization** – The decompressed object is deserialized into memory without type restrictions.

Because BinaryFormatter can instantiate arbitrary objects during deserialization, an attacker can craft a malicious serialized object graph. By embedding dangerous object types (such as a SortedSet<string> with a malicious MulticastDelegate), the attacker can coerce the runtime into executing System.Diagnostics.Process.Start, launching arbitrary commands on the host system.

Exploitation Requirements

- **Prerequisites:**
  - **Valid authenticated credentials for Web Deploy** (low-privilege sufficient).
  - **Network access to the Web Deploy service** (via TCP 8172 for WMSvc or the MsDepSvc port).
- **Attack Complexity:** Low – Exploitation only requires sending a crafted HTTP POST with a malicious SyncOptions header.
- **User Interaction:** None required.

Key Technical Parameters

- **Protocol: HTTP(S)**
  - **Web Management Service (WMSvc)** → TCP 8172 (default)
  - **Web Deploy Agent Service (MsDepSvc)** → Custom (default TCP 80/443 or configurable)
- **Attack Surface:** Internet-facing or internally exposed IIS servers running Web Deploy.
- **Exploit Vector:** Malicious MSDeploy.SyncOptions HTTP header.
- **Vulnerability Class:** Insecure Deserialization (Remote Code Execution).
- **Privilege Requirement:** Authenticated access (low privileges sufficient).
- **Impact:** Full Remote Code Execution on target IIS host.

Technical Analysis

The remote code execution vulnerability in Microsoft Web Deploy (CVE-2025-53772) arises from insecure deserialization of the MSDeploy.SyncOptions HTTP header. When a request is sent to the Web Deploy service, the value of this header undergoes a three-step process: Base64 decoding → GZip decompression → .NET BinaryFormatter deserialization. The critical weakness lies in the final stage. The BinaryFormatter API is inherently unsafe because it can instantiate any serializable object type. In this case, the deserialization routine performs no type checking or validation, allowing attackers to supply a malicious serialized object graph that executes unintended code during reconstruction.

From an attacker's perspective, the exploitation requirements are minimal. Only low-privilege authenticated credentials are necessary, and no user interaction is required. The vulnerable endpoints are:

/msdeploy.axd served by the Web Management Service (WMSvc, default TCP 8172)

MsDepSvc (Web Deploy Agent Service, configurable port, often bound internally on TCP 80/443)

With credentials and network access to either of these services, an attacker can reliably deliver a malicious payload.

The exploitation method follows a straightforward chain. First, the attacker builds a crafted .NET object graph that uses a known gadget chain.Second, the attacker serializes this object with BinaryFormatter, compresses the bytes with GZip, and encodes the result in Base64 to match the expected header format. Third, the payload is delivered inside an HTTP POST request with the header MSDeploy.SyncOptions.

On the server side, the vulnerable code path blindly processes the header. The Base64 blob is decoded, decompressed, and deserialized, at which point the malicious gadget executes. If the target is /msdeploy.axd, the resulting process runs in the IIS Web Management Service (WMSvc) context. If the target is MsDepSvc, the process runs under the configured service account of that Windows service. In either case, child processes like cmd.exe, powershell.exe, or custom binaries will appear as descendants of w3wp.exe (IIS worker process) or MsDepSvc.exe. This creates clear forensic indicators in process trees but can bypass standard IIS logs, which do not capture custom headers.

Several technical markers make this vulnerability dangerous but detectable if defenders know what to look for:

Unusually large or malformed MSDeploy.SyncOptions headers in POST requests.

Base64/GZip-encoded blobs where deployment metadata would normally appear.

Unexpected process creation events on servers, such as cmd.exe or powershell.exe spawned by w3wp.exe or MsDepSvc.exe.

Service account abuse, where a low-privilege deployment account executes system-level commands.

This combination of low attack complexity, reliable gadget chains, and lack of native IIS visibility makes CVE-2025-53772 a high-severity threat. In DevOps environments where Web Deploy is internet-facing, exploitation can provide attackers with direct control over deployment servers and, by extension, the entire CI/CD pipeline.

Potential Impact

Exploitation of CVE-2025-53772 allows an attacker to achieve full remote code execution (RCE) on Microsoft IIS servers running Web Deploy 4.0. Because the vulnerability can be triggered with only low-privilege authenticated credentials and requires no user interaction, the barrier to successful exploitation is low, and the potential damage is severe.

Once exploited, an attacker gains the ability to execute arbitrary system commands under the context of the Web Deploy service. In practice, this enables a wide range of post-exploitation activities, including:

- **Server Compromise and Takeover**

  Attackers can run commands such as cmd.exe or powershell.exe to install additional tools, backdoors, or ransomware, effectively taking full control of the IIS server.

- **Privilege Escalation**

  Although the vulnerability requires low-privilege credentials, successful exploitation often allows attackers to escalate privileges by spawning processes under service accounts or leveraging secondary exploits. In compromised environments, attackers can pivot to domain administrator privileges or higher-value accounts.

- **Deployment Pipeline Compromise**

  Because Web Deploy is a critical DevOps tool used to automate updates and synchronize applications, an attacker controlling this service can tamper with deployment artifacts, application binaries, or configuration files. This threatens the integrity of production applications and undermines the trustworthiness of the entire CI/CD pipeline.

- **Data Exfiltration**

  Arbitrary command execution allows adversaries to access sensitive files stored on the server, harvest credentials from memory or configuration files, and exfiltrate databases or application secrets.

- **Service Disruption and Downtime**

  Attackers could intentionally disable services, overwrite configuration files, or corrupt deployed applications, resulting in outages and downtime for critical web services hosted on IIS.

- **Persistence and Lateral Movement**

  With control of the server, adversaries can establish persistence (e.g., scheduled tasks, registry modifications, malicious services) and then use the compromised IIS host as a foothold to move laterally inside the enterprise network.

From a business perspective, the consequences extend beyond the single vulnerable server. Because Web Deploy often sits at the center of deployment workflows, compromise of this service can cascade into:

- **Loss of application integrity** – attackers inject malicious code into production deployments.
- **Loss of customer trust** – web applications are hijacked to serve malicious content.
- **Regulatory and compliance exposure** – data breaches involving sensitive customer or enterprise data.
- **Operational risk** – outages and costly incident response due to compromised DevOps pipelines.

Indicators of Exploitation

Detecting exploitation of CVE-2025-53772 requires monitoring for unusual patterns in IIS logs, system events, and process activity that deviate from normal Web Deploy operations. Since the malicious payload is injected into an HTTP header (MSDeploy.SyncOptions), which IIS does not log by default, defenders must rely on a combination of network monitoring, endpoint detection, and process analysis.

1. Network and IIS Log Artifacts

- Unusual HTTP POST requests to:
  - **/msdeploy.axd endpoint** (Web Management Service, default TCP 8172)
  - **MsDepSvc service endpoint** (configurable port, often internal)
- Requests containing oversized or malformed MSDeploy.SyncOptions headers with long Base64-encoded blobs.
- High frequency of deployment requests outside normal change windows or from unfamiliar IP addresses.
- IIS logs showing repeated POST activity to /msdeploy.axd without corresponding legitimate deployment events.

2. Process Creation Patterns

- Unexpected child processes spawned by Web Deploy services:
  - **w3wp.exe** (IIS worker process for WMSvc) → launching cmd.exe, powershell.exe, or suspicious executables.
  - **MsDepSvc.exe** (Web Deploy Agent Service) → spawning calc.exe (PoC), or real-world commands like cmd.exe /c whoami or powershell.exe -nop -w hidden.
- **Short-lived processes** (e.g., calc.exe, notepad.exe) used in PoCs as indicators of successful exploitation.
- **Persistence mechanisms** initiated shortly after exploitation, such as new scheduled tasks, registry modifications, or service installations.

3. Windows Event and Sysmon Indicators

- **Sysmon Event ID 1 (Process Creation):** Look for parent–child relationships:
  - **Parent:** w3wp.exe or MsDepSvc.exe
  - **Child:** cmd.exe, powershell.exe, rundll32.exe, or custom binaries.
- **Event ID 4688 (Process Creation):** In Windows Security Logs, watch for suspicious command-line arguments passed to cmd.exe or powershell.exe.
- **Event ID 7045 (Service Installation):** Unexpected services installed by Web Deploy service accounts.

4. Suspicious Command Usage

- **Commands commonly executed post-exploitation:**
  - whoami, ipconfig, net user, net localgroup administrators → used for reconnaissance.
  - powershell -enc <Base64> → used for downloading and executing second-stage payloads.
  - certutil -urlcache -split -f <URL> → often used to fetch malware.

5. File System and Registry Artifacts

- New executables or scripts appearing in temporary directories (C:\Windows\Temp\, %APPDATA%\) shortly after suspicious Web Deploy activity.
- Registry modifications enabling persistence (e.g., Run keys, scheduled tasks) tied to the Web Deploy service account.

Mitigation & Remediation

The critical vulnerability in Microsoft Web Deploy (CVE-2025-53772) demands immediate and layered defensive action. Because a public proof-of-concept (PoC) is available and exploitation requires only low-privilege credentials, organizations should assume that this flaw will be actively targeted. Remediation should combine vendor patching, service hardening, and defense-in-depth strategies.

1.Apply Security Updates

Patch Priority: Microsoft has released Web Deploy 4.0 version 10.0.2001 to address this flaw. This update replaces unsafe BinaryFormatter deserialization with secure logic, closing the exploit path.

Action:

Download and install the patch from the official update channels.

Verify successful patching by checking Web Deploy build version (≥ 10.0.2001).

Risk if Unpatched: Unpatched systems remain vulnerable to authenticated remote code execution and full server compromise.

2.Disable Unnecessary Services

MsDepSvc (Web Deploy Agent Service): If not strictly required, disable or uninstall this service. This immediately removes one of the vulnerable attack surfaces.

/msdeploy.axd Handler (WMSvc): Remove or disable the IIS handler if deployments do not require remote access through the Web Management Service.

3.Restrict Access to Endpoints

Network Controls:

Limit exposure of Web Deploy endpoints (/msdeploy.axd, MsDepSvc) to trusted IP addresses only.

Use VPN tunnels or bastion hosts for administrative access instead of exposing services directly to the internet.

Apply firewall rules to block all inbound connections except from explicitly authorized hosts.

Authentication Hardening:

Enforce multi-factor authentication (MFA) for accounts with deployment privileges.

Apply least privilege to all service and deployment accounts; remove local admin rights wherever possible.

4. Monitor for Exploitation Attempts

IIS and Network Monitoring: Watch for anomalous POST requests to /msdeploy.axd with unusually large MSDeploy.SyncOptions headers.

Process Monitoring: Deploy endpoint detection rules for suspicious child processes (e.g., cmd.exe, powershell.exe) spawned by w3wp.exe or MsDepSvc.exe.

Log Collection: Enable detailed Windows Event Logs and Sysmon to detect process creation, service installation, and registry persistence activities.

5. Strengthen Deployment Infrastructure

Segmentation: Isolate Web Deploy hosts in dedicated network segments with minimal communication paths to other critical systems.

Service Accounts: Run WMSvc and MsDepSvc under restricted service accounts with only the permissions required for deployment.

Configuration Review: Regularly audit Web Deploy configurations and IIS settings to identify unused endpoints, weak access controls, or insecure bindings.

ATHENIAN TECH

Contact Us

www.atheniantech.com