Explanation and Work-around on CSCvt73723 - WebRTC server leaking sessions after large amount of sessions placed on the server

Contents

[Introduction](#)
[Prerequisites](#)
[Requirements](#)
[Components Used](#)
[CSCvt73723 - WebRTC server leaking sessions after large amount of sessions placed on the server](#)
[How do you identify this bug?](#)
[How can you avoid this issue?](#)
[When is this planned to be fixed?](#)
[Related Information](#)

Introduction

This document describes the detection and work-around on Cisco bug id CSCvt73723 around WebRTC server leaking sessions after large amount of sessions placed on the server. That can eventually cause users to unable to log in or join as guest on the WebBridge.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco Meeting Server (CMS) (CallBridge and WebBridge component)

Components Used

The information in this document is based on Cisco Meeting Server and in particular around the WebBridge 2 / CMA WebRTC component. This document does not apply for the new WebBridge 3 / CMS Web app component that has been introduced in version 2.9.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

CSCvt73723 - WebRTC server leaking sessions after large amount of sessions placed on the server

How do you identify this bug?

The symptom from an end user perspective is once they have hit the hard limit and no further users can join a meeting. In the logs, spotting the webbridge stats (as per this FAQ) are hitting 149 does **NOT** necessarily imply these are all leaked sessions. This just means that the Web Bridge has hit its hard limit and no new connections are allowed.

"webbridge": INFO : [DEBUGGING] Stats 149, c:3477, d:3170

Calculating how many of these are leaked sessions is a little more complicated and can be done if you are **NOT** using the CMA desktop client or iOS client. From version 2.8, the Call Bridge does report every 5 minutes the number of CMA sessions (CMA WebRTC + CMA desktop client + CMA iOS client). Note this is reported as the "CMA": "X/Y" where X is the current number of active CMA sessions and Y is the peak in the last 5 minutes.

INFO : STATS: {"callLegsPS": 1, "callLegs": "20/24", "CMA": "14/17", "sip": {"std": "0/1", "peer": "6/6"}}

Just because a Call Bridge is reporting 14 current session does not mean the co-located Web Bridge is also reporting 14 sessions. This mapping is 1:1 on a single combined server but in a clustered deployment a Web Bridge session can instantiate a call on a different Call Bridge (especially when load balancing is enabled - which it is by default for CMA).

Therefore, in order to calculate the total number of leaked sessions in a deployment you do need the combined active sessions from ALL the Web Bridge stats and compare this with the combined CMA Call Bridge stats that are reported.

How can you avoid this issue?

Depending on how often your deployment hits this situation (once every couple of days or once every couple of weeks), you must be advised to restart their Web Bridge(s) which does clear out any leaked sessions and reset the active sessions count to 0. Understandably this can be tedious if this becomes a daily chore hence why this task can be facilitated with a script available as per the code block.

```
######################################################################
#### Cisco Meeting Server ####
#### Webbridge restart ####
#### Workaround for CSCvt73723 ####
#### feedback: willwoo@cisco.com ####
######################################################################
#----------------------------------------------------------
# ---------- DISCLAIMER ----------
#----------------------------------------------------------
# Please note this script is NOT maintained or supported by Cisco.
# This is to be run at entirely your own risk.
# This script is not intended for redistribution
# Tested with python 3.7.4
#----------------------------------------------------------
#----------------------------------------------------------
# ---------- Libraries to import ----------
#----------------------------------------------------------
import paramiko
import time
import datetime
#----------------------------------------------------------
```

```
#-------------------------------------------------------------
# ---------- Deployment parameters to change ----------
#-------------------------------------------------------------
# WB Inventory - just extend or modify the below to match your deployment requirements.
# Enter the MMP IP of the server (can differ from interface webbridge service is running)
webbridges ={1:"127.0.0.1",2:"127.0.0.1",3:"127.0.0.1",4:"127.0.0.1"}

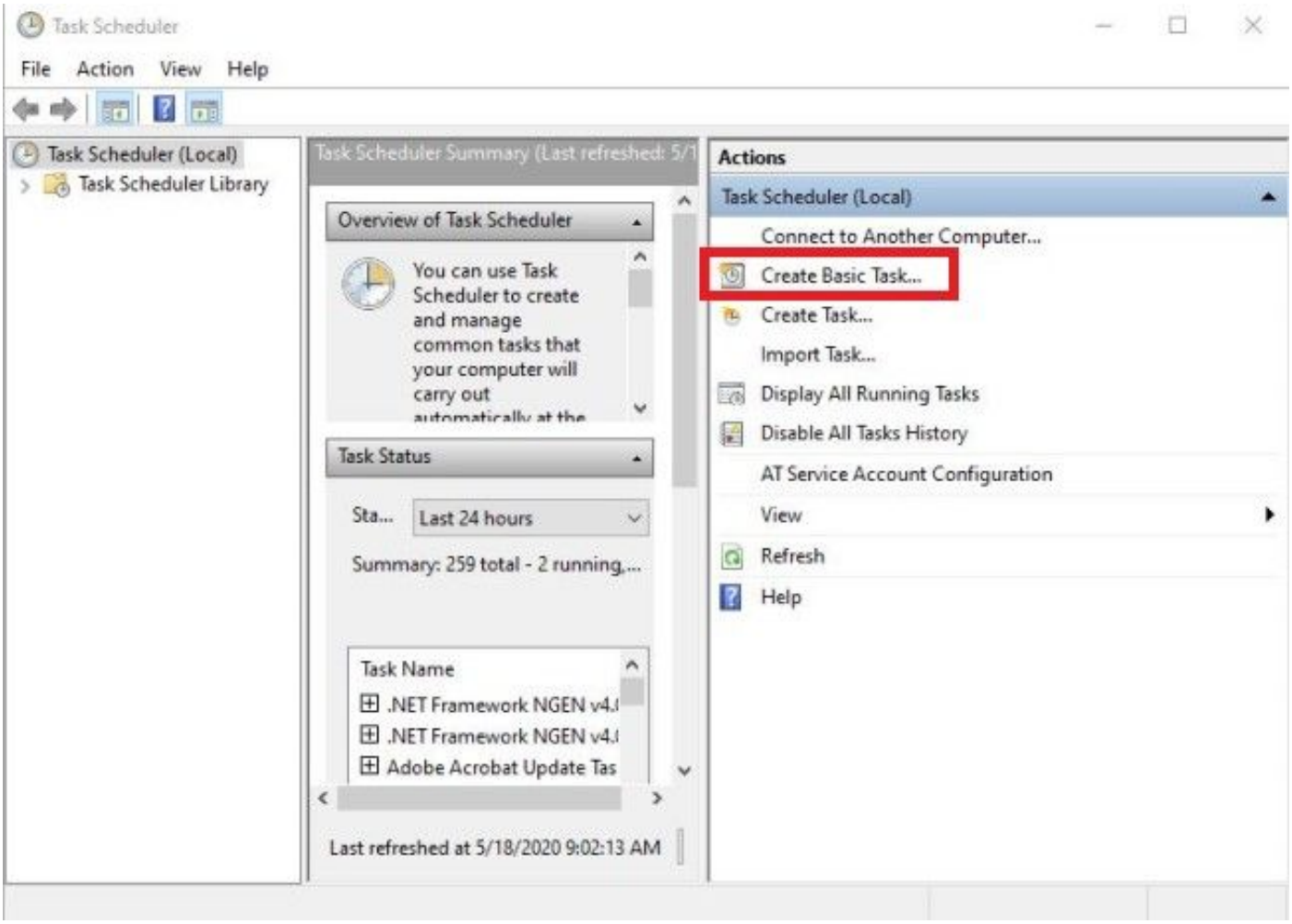
mmp_username = "admin" # MMP username
mmp_password = "password" # MMP password
#-------------------------------------------------------------
def mmp_webbridge_restart(mmp_address,uname,pword):
conn = paramiko.SSHClient()
conn.set_missing_host_key_policy(paramiko.AutoAddPolicy())
try:
conn.connect(mmp_address, 22, uname, pword)
stdin, stdout, stderr = conn.exec_command('webbridge restart')
time.sleep(1)
conn.close()
print_log_message('Webbridge on server: ' + mmp_address + ' restarted successfully')
except Exception as error:
print_log_message('Failed to restart webbridge on server ' + mmp_address + '. Error:')
print_log_message(str(error))
pass
def print_log_message(message):
time_stamp = datetime.datetime.now(datetime.timezone.utc)
time_stamp = str(time_stamp)
file = open('webbridge_restart_logs.txt', 'a')
file.write(time_stamp + " " + message + "\n")
file.close()
if __name__ == '__main__':
for wb in webbridges:
mmp_webbridge_restart(webbridges[wb], mmp_username, mmp_password)
#####################################################################
```

The script requires some small edits (the credentials on line 29-30 and IP addresses of the Web Bridges in the deployment on line 27) and must **ONLY** be run when there is no expected load or during a maintenance window. The script does **not** check for active sessions and simply performs the 'webbridge restart' command on all servers listed which does terminate any active WebRTC session.

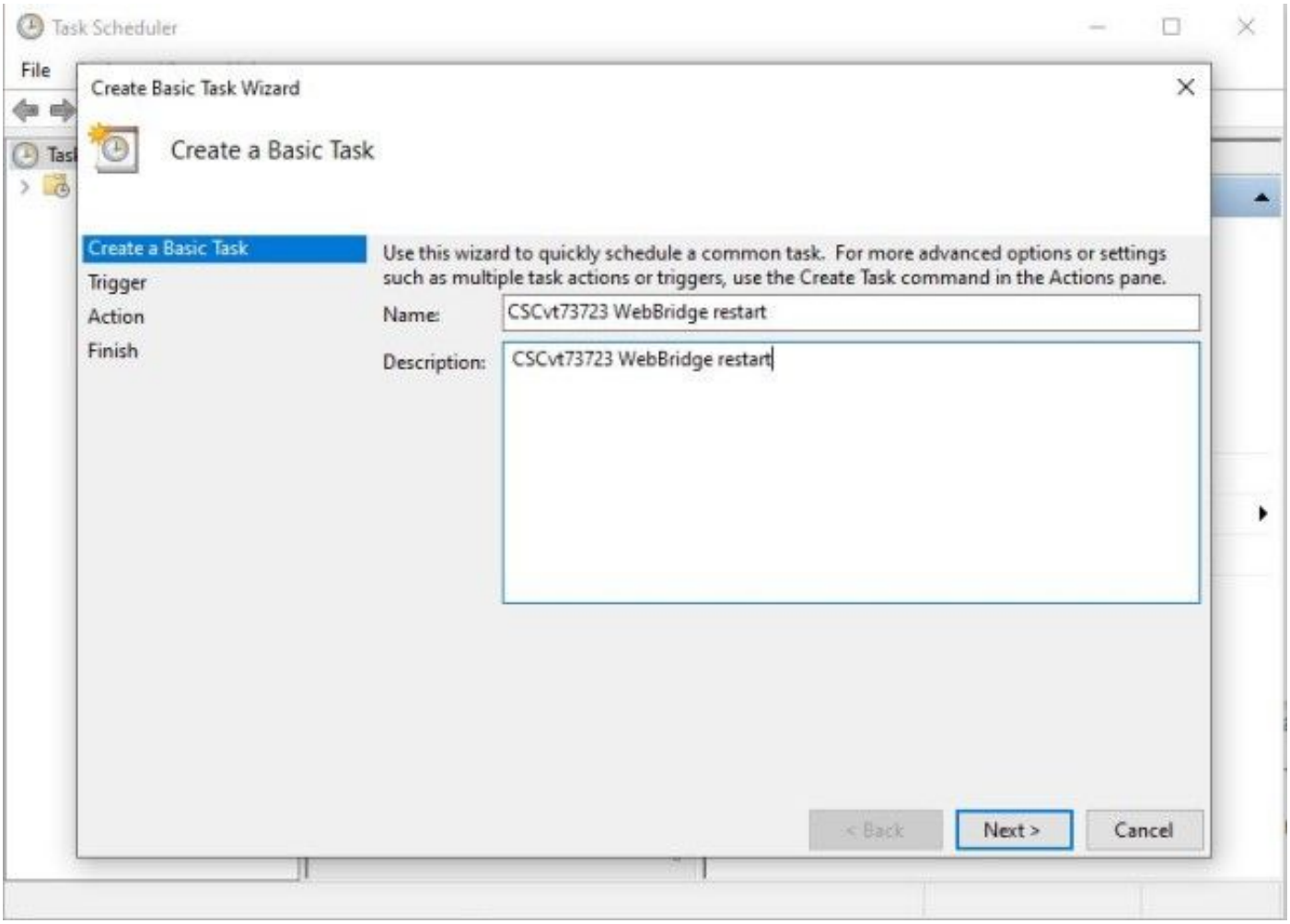
To automate this script, it can be done by setting up a cron job or on a Windows 10 PC with Task Scheduler. Assuming the Win 10 PC has Python 3.4+ installed they can follow these steps:

1. Open Task Scheduler

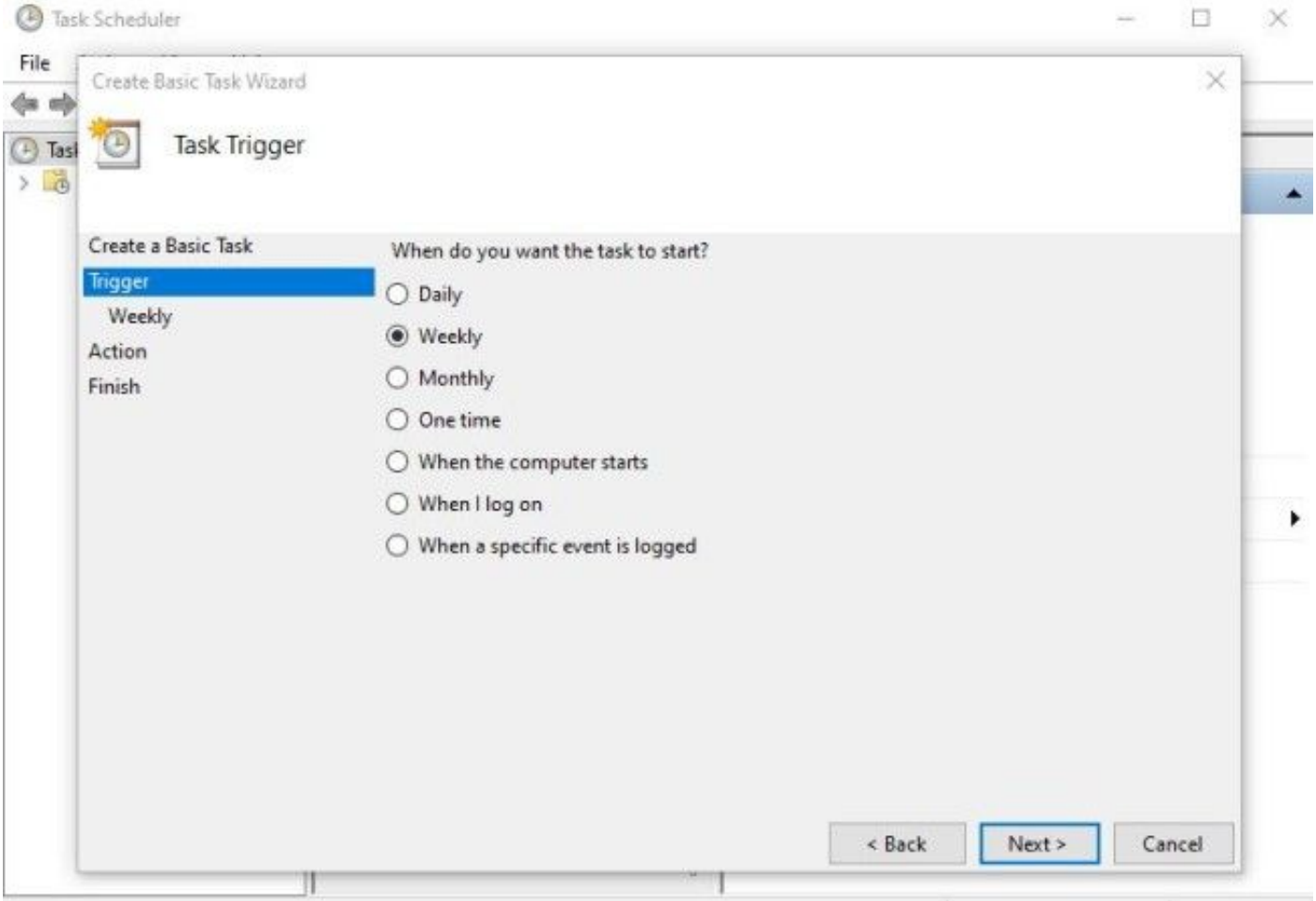
2. Select 'Create Basic Task...'

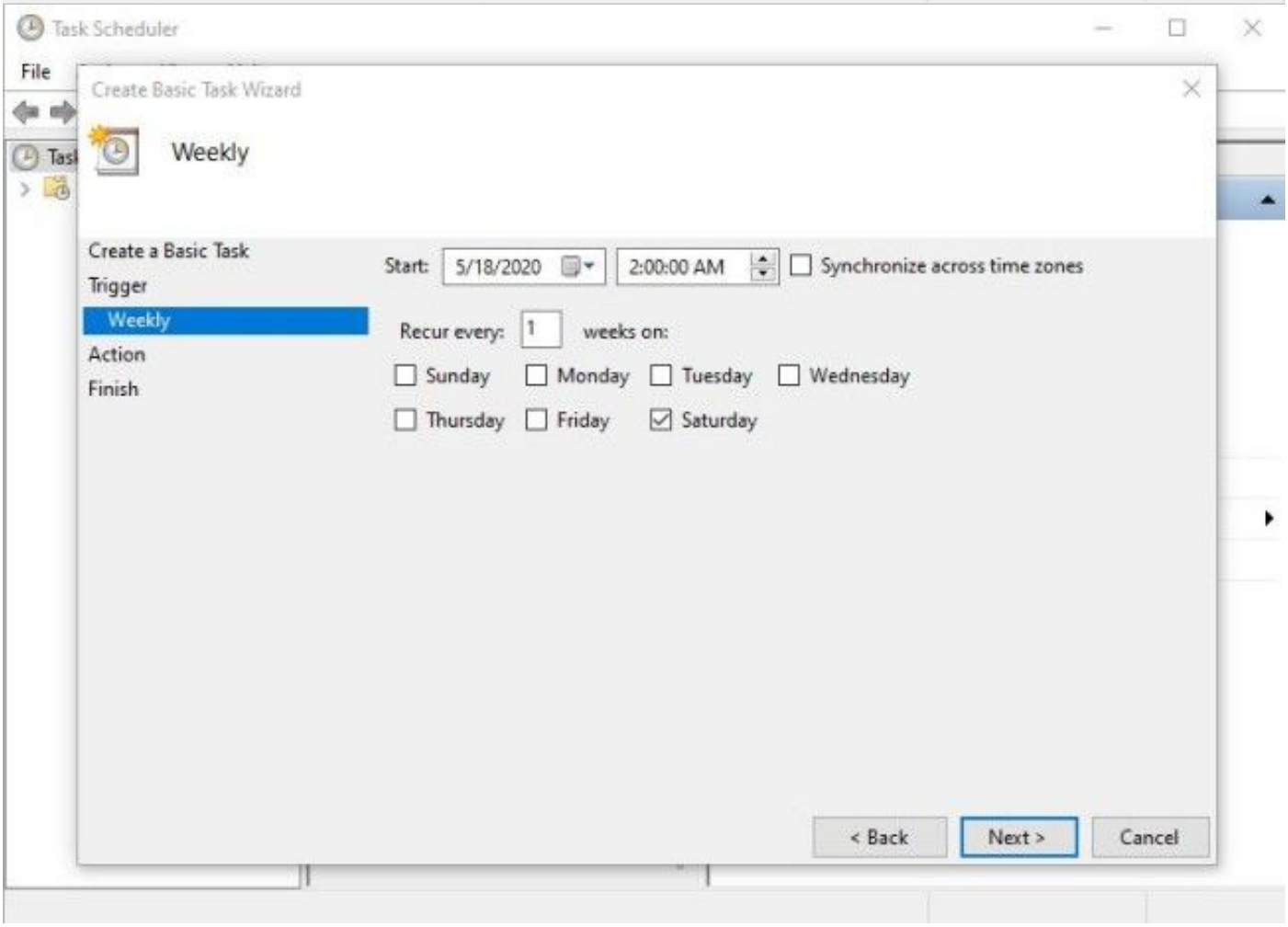


2.1 Enter a Name / Description for this task

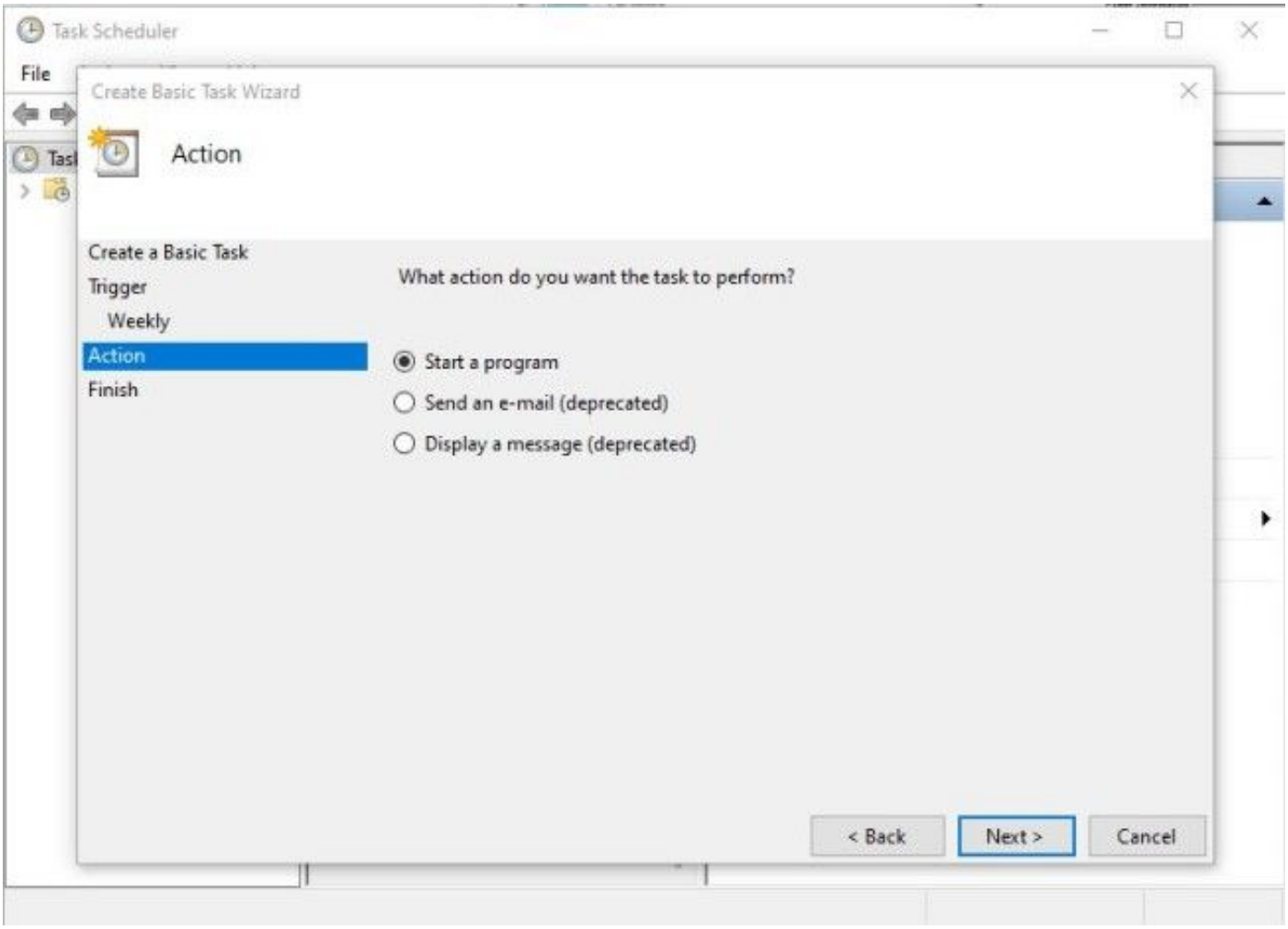


2.2 Select what frequency and times you want to run this task (recommended only to be during off-peak times, here shown for every Saturday at 2AM)


Task Scheduler
File
Create Basic Task Wizard
Task Trigger
Create a Basic Task
Trigger
Weekly
Action
Finish
When do you want the task to start?
Daily
Weekly
Monthly
One time
When the computer starts
When I log on
When a specific event is logged
< Back
Next >
Cancel


Task Scheduler
File
Create Basic Task Wizard
Weekly
Create a Basic Task
Trigger
Weekly
Action
Finish
Start: 5/18/2020 2:00:00 AM
Synchronize across time zones
Recur every: 1 weeks on:
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
< Back
Next >
Cancel

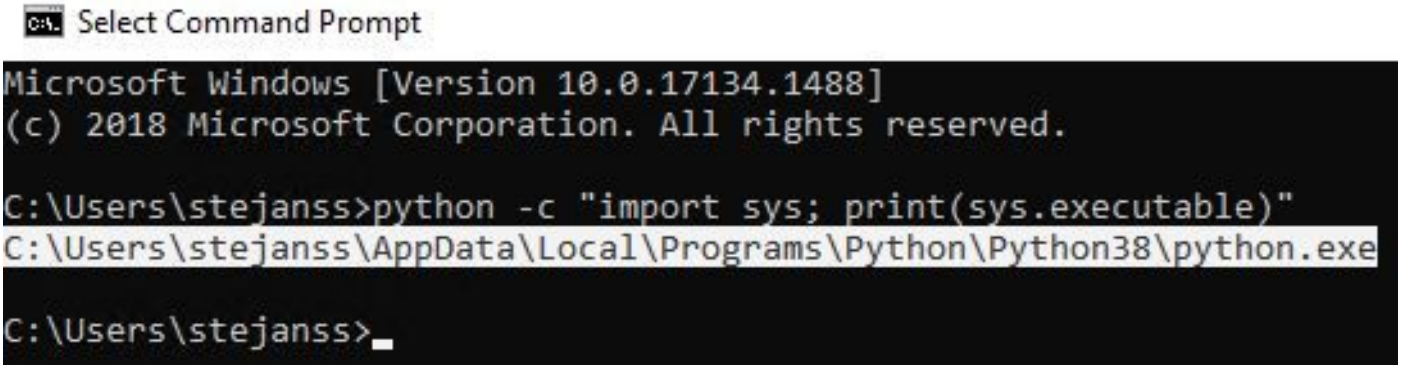
2.3 Action to perform, select: 'Start a program'



2.4 Action:

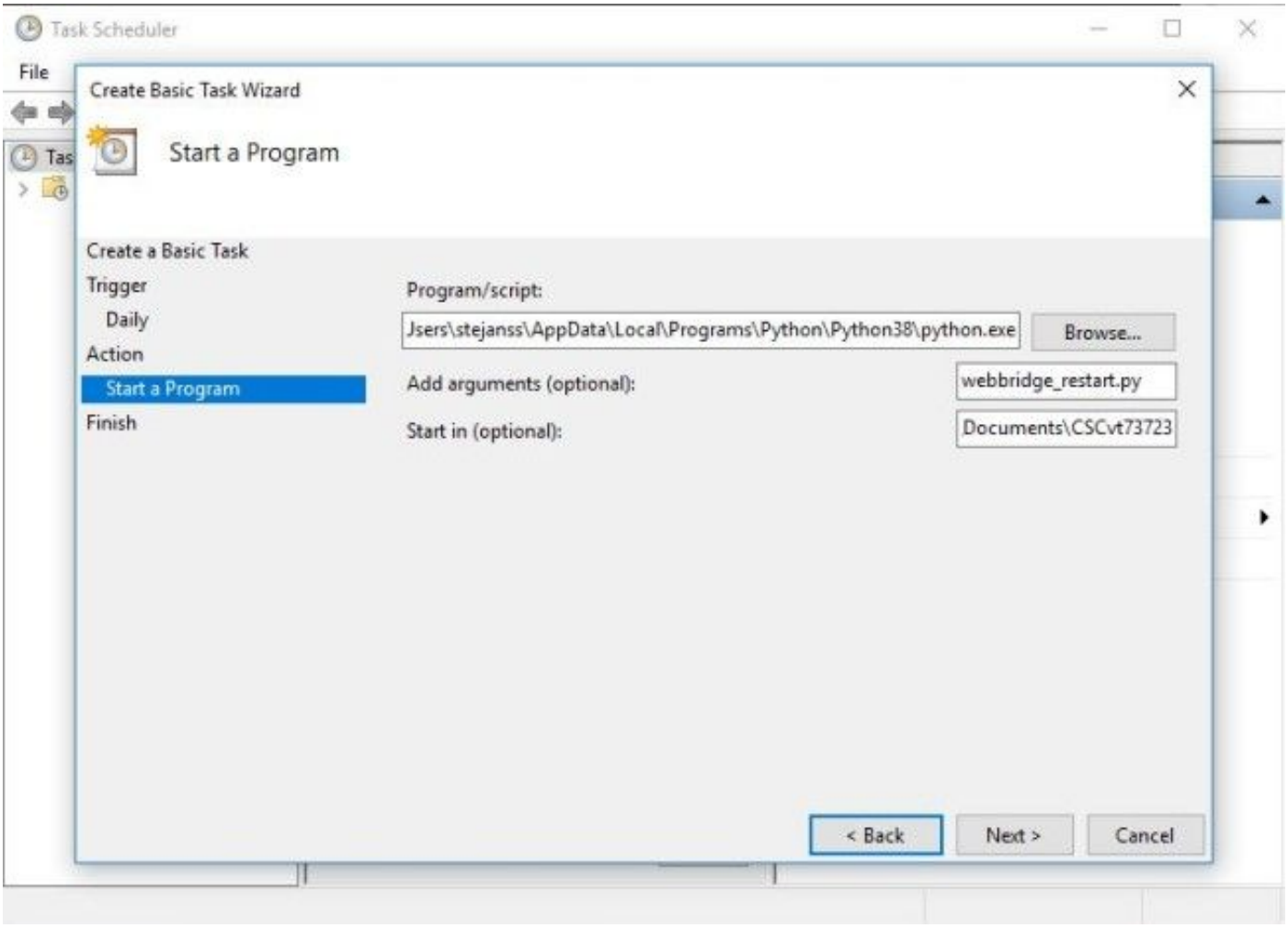
* Program / Script: **C:\<path to python.exe>**

(if you don't know the path to python.exe, it can be found by going to cmd and typing: *python -c "import sys; print(sys.executable)")*



* Add arguments (optional): **webbridge_restart.py** (or name of python script)

* Start in (optional): **C:\<path to webbridge_restart.py>**



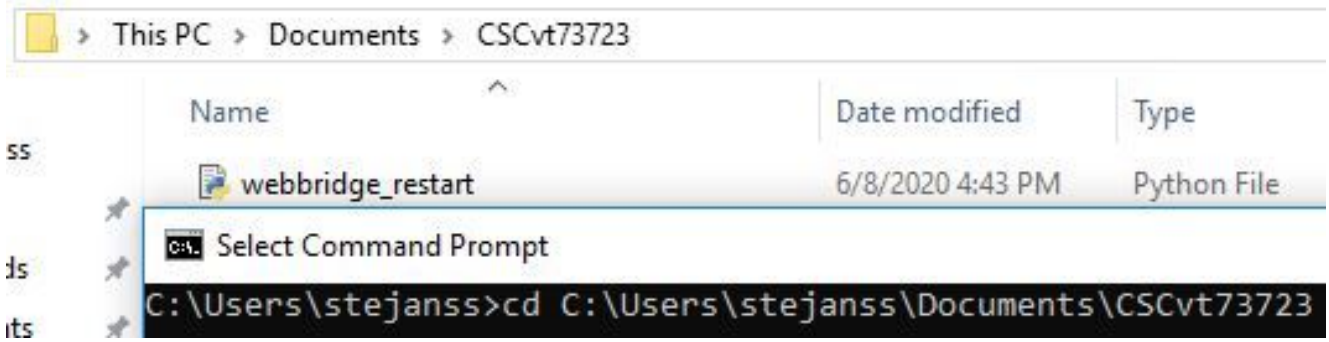
Note that computer that's running the cron job must be able to access the MMP of the CMS servers configured. After the script has run, it creates a webbridge_restart_logs.txt file that contains details on the restarts of the different WebBridges as well as any potential failures. An example is shown with one successful connection to 10.48.79.194 and one failing one to 127.0.0.1 (as being the loopback address of the PC actually).

2020-06-08 14:53:18.149915+00:00 Webbridge on server: 10.48.79.194 restarted successfully 2020-06-08 14:53:19.165543+00:00 Failed to restart webbridge on server 127.0.0.1. Error: 2020-06-08 14:53:19.165543+00:00 [Errno None] Unable to connect to port 22 on 127.0.0.1

How to test that the script works fine?

If you have Python installed the PC where you inted to run the script from, you can run on it manually first with the next steps:

1. Open cmd and browse to the location of the script with the '**cd**' command



2. Run the python file with the command '**python webbridge_restart.py**'

```
C:\Users\stejanss\Documents\CSCvt73723>python webbridge_restart.py
Traceback (most recent call last):
  File "webbridge_restart.py", line 18, in <module>
    import paramiko
ModuleNotFoundError: No module named 'paramiko'
```

3. In case you see an error indicating that the 'paramiko' module is not installed, you need to install some extra library with the command '**pip install paramiko**'

```
C:\Users\stejanss\Documents\CSCvt73723>pip install paramiko
Collecting paramiko
  Downloading https://files.pythonhosted.org/packages/06/1e/1e08b
aaaf6c3d3df1459fd85f0e7d2d6aa916f33958f151ee1ecc9800971/paramiko-
2.7.1-py2.py3-none-any.whl (206kB)
     |                                | 215kB 1.3MB/s
Collecting cryptography>=2.5 (from paramiko)
```

4. Once completed, you can run the script again with '**python webbridge_restart.py**' (NOTE: this restarts the webbridge and causes current ongoing WebRTC connections to disconnect)

   If it has run successfully, you can check the outcome of it in the webbridge_restart_logs.txt file.

When is this planned to be fixed?

This is not a new bug and there is no plan on fixing this on the Web Bridge 2 / CMA WebRTC. The new Web Bridge 3 / CMS web app (available from 2.9 onwards) are not affected by this bug as it has been completely redesigned. Customers who are heavily impacted by this must consider moving to the new CMS web app (although note this is not yet feature parity with Web Bridge 2 in the 2.9 release. Check the CMS 2.9 and cms web app release notes for full details on this.)

Related Information

- **Defect notes: https://bst.cloudapps.cisco.com/bugsearch/bug/CSCvt73723**
- **Release notes: https://www.cisco.com/c/en/us/support/conferencing/meeting-server/products-release-notes-list.html**
- **FAQ https://meeting-infohub.cisco.com/faq/content/43/450/en/how-do-you-check-the-number-of-active-sessions-on-a-web-bridge.html**